COMPILER
Vol. 12, No. 2, November 2023, pp. 117~126
ISSN: 2252-2403 (Print), ISSN: 2549-2403 (Online)
Accredited Fourth Grade by Kemenristekdikti, Decree No:200/M/KPT/2020
DOI: 10.28989/compiler.v12i2.1915

Combatting Heart Diseases: Advanced Predictions Using Optimized DNN Architecture

Mochammad Abdul Azis[1*], Sumarna[2]
[1]Department of Information Technology, Universitas Bina Sarana Informatika, Indonesia
[2]Department of Information Technology, Universitas Nusa Mandiri, Indonesia

Article Info

Article history:

Received October 26, 2023
Accepted November 14, 2023
Published November 30, 2023

Keywords:

DNN
Deep Learning
Heart Disease
Classification

ABSTRACT

Heart disease has become a global health issue and is recorded as one of the primary causes of death in many countries. In this modern era, with rapid technological advancements and shifting lifestyles, numerous factors contribute to the increasing prevalence of heart diseases. These range from dietary habits, lack of physical activity, and stress, to genetic factors. Given the complexity of this ailment, information technology plays a crucial role in providing innovative solutions. One of them is predicting the risk of heart disease, enabling more targeted early prevention and treatment interventions. Correct data analysis is pivotal in making predictions. However, a common challenge often encountered is the imbalance in data classes, which can result in a predictive model being biased. This is certainly detrimental, especially in the context of predicting strokes, where prediction accuracy can mean the difference between life and death. In this research, our focus was on developing a Deep Neural Network (DNN) Architecture model. This model aims to offer more accurate predictions by considering data complexities. By optimizing several key parameters, such as the type of optimizer, learning rate, and the number of epochs, we strived to achieve the model's best performance. Specifically, we selected Adagrad as the optimizer, set the learning rate at 0.01, and employed a total of 100 epochs in its training. The results obtained from this research are quite promising. The optimized DNN model displayed an accuracy score of 0.92, a precision of 0.92, a recall of 0.95, and an f-measure of 0.93. This indicates that with the right approach and meticulous optimization, technology can be a highly valuable tool in combatting heart diseases.

Corresponding Author:

Mochammad Abdul Azis,
Department of Information Technology,
Universitas Bina Sarana Informatika,
Jl. Kramat Raya No.98, RT.2/RW.9, Kwitang, Kec. Senen, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10450.
Email: *mochamad.mmz@bsi.ac.id

1. INTRODUCTION

The current global health crisis is characterized by a high rate of deaths due to heart attacks, with developing countries in Asia and Africa being particularly affected owing to delays in assessing the severity of heart attacks [1]. Despite collecting essential diagnostic data in medical practice, there is a significant gap in effectively using these datasets for the early detection of heart attacks, which is crucial for prevention [2]. This research aims to address this gap by leveraging real-life datasets to accurately and timely predict heart attacks. Although current data analysis and mining techniques are available, they are not fully utilized for proactive heart disease prediction. Known risk factors, including cholesterol levels, blood pressure, lifestyle choices, and other cardiovascular conditions, have underutilized predictive capabilities through AI [3]. The application of artificial intelligence, particularly machine learning (ML) and deep learning (DL) is crucial in this context. In previous research conducted by Kumar et al., titled “Analysis and Prediction of Cardiovascular Disease using Machine Learning Classifiers”, the study focused on determining which classifier achieved higher precision and AUC ROC scores. The findings indicated that the Random Forest classifier achieved a higher precision of 85%, an AUC ROC score of 0.8675, and an execution time of 1.09 seconds [4]. In another prior study conducted

by Saiful Islam et al., titled “Cardiovascular Disease Forecast using Machine Learning Paradigms”, the research utilized 301 sample data points with 12 clinical attributes. The classification algorithms used to predict heart disease included Logistic Regression, Decision Tree, SVM, and Naive Bayes, with Logistic Regression achieving an accuracy of 86.25% [5]. However, Deep Neural Networks (DNN), while powerful, encounter challenges such as complexity that leads to overfitting, the necessity for large datasets, limited transparency, and high computational demands. These challenges impede their practical application in medical diagnostics [6]. DNNs also face issues like the need for extensive hyperparameter tuning and their "black box" nature, which complicates their interpretability in clinical settings. Despite these limitations, DNNs offer significant advantages, including their ability to learn high-level features from complex data and process large-scale datasets, which is vital for medical applications. Their adaptability and the potential for high predictive accuracy make them a promising tool in medical diagnostics, especially in detecting subtle patterns indicative of early heart attacks. The study strives to surmount these challenges by developing an optimized deep learning model that focuses on the Neural Network (NN) algorithm. Its objectives include enhancing predictive accuracy, reducing overfitting through regularization and cross-validation, efficiently utilizing data, improving model transparency for medical practitioners, and optimizing computational resources [7][8]. By refining this model and harnessing AI's capabilities, the study aspires to exceed current accuracy benchmarks and make a substantial contribution to the early detection and prevention of heart attacks, thereby saving lives. This research not only furthers scientific knowledge in AI and healthcare but also significantly impacts patient health outcomes, particularly in the early detection of heart attacks.

2. RESEARCH METHOD

2.1. Data Cleaning

The objective of data cleaning is to remove noise, inconsistent data, and errors in the dataset. One form of noise is the occurrence of missing values. One technique that can be used to address missing values is by performing imputation. Several imputation approaches are explained as follows [9]:

1) Simple Regression Imputation
   Simple Regression Imputation uses the information available in the data set rather than mean substitution to produce the values used for imputation. Simple regression involves creating a quadratic regression equation, where the missing variable observations serve as the dependent variable and relevant variables in the data set are used to predict the missing values.
2) Regression Imputation with Added Error Term
   Regression imputation with an added error uses a procedure that is almost the same as simple regression to predict Ymiss [10]. The mean imputation, regression imputation and stochastic regression approaches are carried out based on the equation:

$$〖v\_i〗^*=a+ X\_i\, b+ 〖e\_i〗^*, i=1,\ldots N\_m, \quad (1)$$

   Where vi* is the imputed value for the missing response on variable v for example case i, Xi is the K-column row vector of observations on K regression predictors for case i in the imputation model, b is the K-order column vector of the corresponding estimated regression coefficients with variables in X. ei* is the estimated residual from the regression of v on X and Nm is the number of tuples that need to be imputed.

2.2. Feature Encoding

Feature encoding is the process of changing the categorical value of a feature in the form of a label into numerical form. Label encoding is a very simple way to convert categorical values to numeric values. Label encoding is simply assigning an integer value to every possible value and is different from categorical variables [11]. This approach is very simple and involves converting each value in a column to a number. Label encoding refers to converting a label into a numeric form so to convert it into a machine-readable form. This is an important preprocessing step for structured datasets in supervised learning. Table 2.2 shows an illustration of the Label Encoding results:

Table 1. *Illustration of Label Encoding*

ChestPainType	*Label Encoding*
NAP	2
ATA	1
ASY	0

2.3. Feature Scaling

Feature scaling is done to change the value vector in the feature into a format that is more suitable for training. Some of the most commonly used feature scaling techniques are Normalizer, MinMaxScaler, StandardScaler, and RobustScaler [12].
The normalizer scales the data for each sample value to the unit norm. The transformed value for feature x is:

$$z= x\_i/\sqrt{(〚x\_i〛^2+〚y\_i〛^2+〚z\_i〛^2)} \quad (2)$$

Where xi, yi and zi are the values of features x, y and z.
MinMaxScaler scales the data so that all values in the dataset are between 0 and 1.

$$t= x\_min/(x\_max- x\_min) \quad (3)$$

Where, t is the transformation result value
x is the original value
x_{min} and x_{max} are the minimum and maximum values on feature x.

StandardScaler transforms the dataset so that the mean value of the resulting distribution is zero and the standard deviation is one. The transformed value is obtained by subtracting the mean value from the original value and dividing by the standard deviation. The formula given below is used for transformation.

$$z=(x-\mu)/\sigma \quad (4)$$

Where, z is the feature value resulting from the transformation
x is the original value
μ is the average value
σ is the standard deviation value

RobustScaler removes the median and scales the data according to a quartile range that ranges from the 25th quartile to the 75th quartile. The transformed value of the data set is relatively larger than the previous scalar. The transformed data value is between the range [−2, 3]. The result is like minmaxscaler but uses the inter quartile range instead of min max. The formula for feature scaling with RobustScaler is as follows:

$$t= (x\_i-Q\_1(x))/(Q\_3(x)-Q\_1(x)) \quad (5)$$

Where, t is the feature value resulting from the transformation
x is the original value
$Q_1(x)$ and $Q_3(x)$ is the inter quartile range

2.4. Confusion Matrix

Confusion Matrix is a tabular visualization of ground truth labels versus model predictions. Each row of the confusion matrix represents an instance in the predicted class and each column represents an instance in the actual class. The Confusion Matrix is not a performance metric, but a kind of baseline against which other metrics evaluate results.

Confusion Matrix is one way to see the performance of a classifier/supervised learning [13]. Confusion Matrix can provide accuracy values for algorithm validation on existing datasets [14]. The first performance metric is the confusion matrix, which is a standard method for presenting the number of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) in a more visual way [15]. The Confusion Matrix is shown by a table, where each column in the table represents the predicted class with the actual class or actual class. For two-class classification (binary classification), the confusion matrix is displays as presented in Table 2 [16].

Table 2. *Confusion Matrix*

Predict	*Actual*	
	Positive	*Negative*
Positive	*TP* (*True Positive*)	*FP* (*False Positive*)
Negative	*FN* (*False Negative*)	*TN* (*True Negative*)

True Positives shows the model prediction results for the number of actual positive classes that were correctly predicted as positive classes. True Negative shows the model prediction results for the number of actual negative classes that are correctly predicted as negative classes. False Positives shows the model prediction results for the number of actual negative classes that were incorrectly predicted as positive classes. False Negatives shows the model prediction results for the number of actual positive classes that were incorrectly predicted as negative classes[17]. Based on this confusion matrix, other performance metrics can be calculated and interpreted.

3. RESULTS AND ANALYSIS

The research starts from identifying the problem object to evaluating the proposed DNN architecture model based on validation accuracy, precision, recall and f1-score.

3.1. Dataset Analysis

The dataset used for this research is secondary data or public data taken from Kaggle. The dataset has 12 features and 918 instances with a file size of 36 kB. All features contained in this dataset are electronic records from patients, mainly based on the patient's basic physiological data and historical disease. The dataset that has been taken from Kaggle is then saved to Google Drive. The characteristics of the dataset have four binary attributes, namely Sex, FastingBS, ExerciseAngina and HeartDisease, seven attributes in categorical form, namely Sex, ChestPainType, FastingBS, RestingECG, ExerciseAngina, ST_Slope and HeartDisease, also seven attributes in numerical form, namely Age, RestingBP, Cholesterol, FastingBS, MaxHR, Oldpeak, and HeartDisease. Attributes are binary, some are numeric and some are text. The dataset attributes used and their explanations are presented in Table 3.

Table 3. *Description of the Dataset Attribute*

Attribute	Description
Age	Patient age
Sex	Patient gender (Male, Female)
ChestPainType	Types of chest pain (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic)
RestingBP	Blood pressure
Cholesterol	Cholesterol
FastingBS	Fasting blood sugar (1: if FastingBS > 120 mg/dl, 0: otherwise)
RestingECG	Electrocardiogram results (Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria)
MaxHR	Maximum heart rate
ExerciseAngina	Exercise-induced angina (Y: Yes, N: No)
Oldpeak	oldpeak = ST [Numerical values are measured in depression]
ST_Slope	Peak exercise ST segment slope (Up: upsloping, Flat: flat, Down: downsloping)
HeartDisease	1 – Heart Disease 0 – Normal

The first ten instances of the dataset are presented in Table 4. The table informs the initial characteristics

of the dataset that there is data that has values in numeric and text form, some are in Categorical text form and some are in Categorical numeric form. Besides that, it also provides information that the data scale for some attributes is different from others.

Table 4. *Dataset*

Age	Sex	ChestPainType	RestingBP	....	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	....	0	Up	0
49	F	NAP	160	....	1	Flat	1
37	M	ATA	130	....	0	Up	0
48	F	ASY	138	....	1.5	Flat	1
54	M	NAP	150	....	0	Up	0
39	M	NAP	120	....	0	Up	0
45	F	ATA	130	....	0	Up	0
54	M	ATA	110	....	0	Up	0
37	M	ASY	140	....	1.5	Flat	1
48	F	ATA	120	....	0	Up	0

3.2. Research Stages

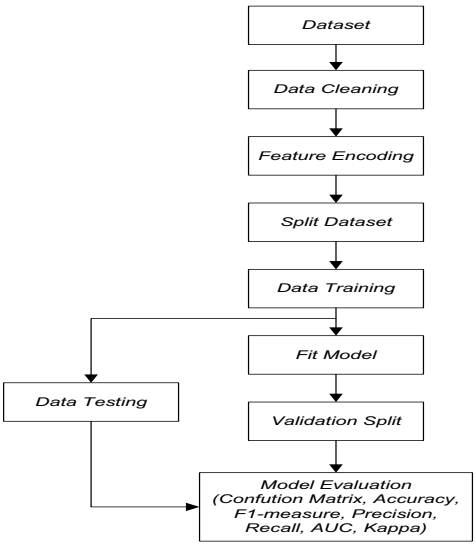

Figure 1. Research Stages

The research was carried out according to the stages as presented in Figure 1. The research starts from identifying the problem object to evaluating the proposed DNN architecture. An explanation of each stage is as follows.

3.3. Feature Importance

The features contained in the dataset have different levels of influence on heart disease. Therefore, it is necessary to use machine learning algorithms to identify the most important features to help determine the occurrence of heart disease from these features. In this way, predictions of heart disease can be made more accurate. Of the eleven predictor features, there is no feature that has a strong influence on heart disease as shown in Table 6 regarding the important coefficient value of each feature. However, there are three features that have a dominant or higher influence than other features. These three features are st_slope, oldpeak and chestpaintype which are identified using Random Forest Feature Importance.

Table 6. Feature Important Coefficient

Feature	Important Coefficient
ST_Slope	0.221342
Oldpeak	0.127714
ChestPainType	0.123113
Cholesterol	0.115153
MaxHR	0.110152
RestingBP	0.080918
ExerciseAngina	0.077147
Age	0.072386
RestingECG	0.026561
Sex	0.026553
FastingBS	0.018960

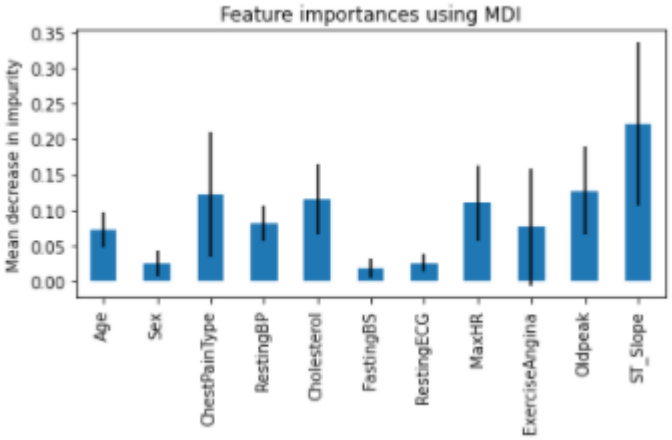

Figure 2. Feature Importance for Random Forest

3.4. Missing Values

Table 7. Missing Values

Feature	Missing Values
Age	0
Sex	0
ChestPainType	0
RestingBP	0
Cholesterol	0
FastingBS	0
RestingECG	0
MaxHR	0
ExerciseAngina	0
Oldpeak	0
ST_Slope	0
HeartDisease	0

Based on observations of the dataset for each feature, no missing values were found, as evidenced by each feature having a value of 0, which indicates that the feature is clean of empty data. Therefore, the dataset used no longer has missing values as presented in Table 7.

3.5. Imbalance Class

The dataset has two target classes with a composition of class 0 (normal) as many as 410 instances (45%) and class 1 (heart disease) as many as 508 instances (55%). Figure 3 shows that the two target classes are not balanced (high imbalance class).

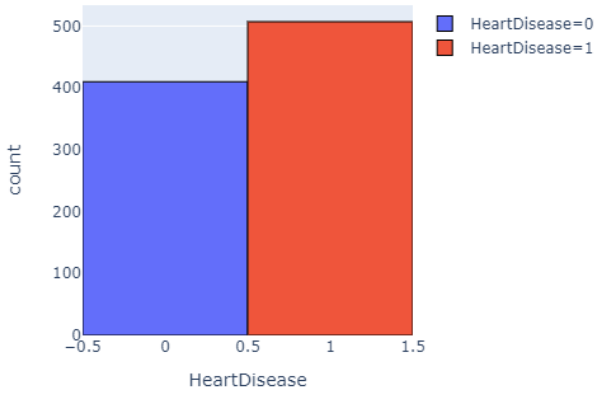

Figure 3. Class Target Distribution

3.6. Experiment Results

Experiments on each model were carried out in stages based on the Split Dataset technique implemented. Testing of hyperparameters was carried out using 4 variations of hidden layers, 2 variations of activation functions, namely in each hidden layer the ReLU activation function was used and in the output layer the Sigmoid activation function was used. Then using the AdaGrad optimizer, and 0.01 variation in learning rate. This model was trained for 100 epochs and a batch size of 50. The experimental results in the form of a Confusion Matrix are in the attachment. The recapitulation of experimental results is presented in the form of a comparison table for each performance metric, namely Accuracy, Recall, Precision, F1-Score and AUC value based on the optimization implemented.

In Table 8. are the Accuracy score results based on AdaGrad optimization. The data in this table can be analyzed as the Accuracy score with AdaGrad optimization, the number of epochs is 100 and the learning rate optimization is 0.01. The accuracy score obtained was 91.85%.

Table 8. Accuracy Score AdaGrad

Optimization	*Epoch*	*LR*	*Accuracy*
AdaGrad	100	0.01	91.85%

In Table 9. are the results of the AdaGrad optimization recall score and the number of epochs is 100. For setting the AdaGrad optimization learning rate value of 0.01, the recall score obtained is 95.00%.

Table 9. Recall Score AdaGrad

Optimization	*Epoch*	*LR*	*Recall*
AdaGrad	100	0.01	95,00%

In Table 10, the results of the AdaGrad optimization precision score with the number of epochs are 100. For setting the learning rate value for AdaGrad optimization, it is 0.01. The precision score obtained was 92.00%.

Table 10. Precision Score AdaGrad

Optimization	*Epoch*	*LR*	*Precision*
AdaGrad	100	0.01	92,00%

In Table 11. F1-score results based on AdaGrad optimization. The data in the table can be analyzed that the score obtained from AdaGrad optimization and the number of epochs is 100. To set the AdaGrad optimization learning rate value of 0.01 F1-score, the AdaGrad optimization is obtained at 93.00%.

Table 11. F1-Score AdaGrad

Optimization	*Epoch*	*LR*	F1-Score
AdaGrad	100	0.01	93,00%

In Table 12, the AUC value results based on AdaGrad optimization get an AUC value of 0.829. For AdaGrad's optimization learning rate it is 0.01. This AUC value is greatly influenced by the True Positive Rate and False Positive Rate.

Table 12. AUC AdaGrad

Optimization	*Epoch*	*LR*	AUC
AdaGrad	100	0.01	0.829

Based on analysis of experimental results from AdaGrad optimization. The architectural parameters of the DNN model are as presented in Table 13. with a total of 11 input neurons according to the feature dataset. The number of hidden layers is 4 with the number of neurons in each layer being 32, 64, 128, 256 units. The activation function for the input layer and hidden layer uses ReLU, while the output layer uses Sigmoid. The optimization used by AdaGrad is with a learning rate of 0.01.

Table 11. Proposed DNN Architecture Parameters

Parameter	**Parameter Value**
Input Neuron	11
Hidden Layer	5
Hidden Neuron	32,64,128,256
Epoch	100
Batch Size	50
Input and Hidden Layer Activation Function	*ReLU*
Output Layer Activation Function	*Sigmoid*
Optimization	AdaGrad (lr = 0.01)

In Table 12, the analysis results of the score for each performance metric are presented. To facilitate comparison with the literature studies referenced in this research, in addition to using the accuracy score, a combination of precision and recall scores is also employed. This is achieved by using the F1-score or f-measure metric, which represents the harmonic mean between precision and recall.

Table 12. Score Metric Performance

Metric Performance	**Score**
Accuracy	91.85%
Precision	92,00%
Recall	95,00%
F1-Score	93,00%
AUC	0.829

4. CONCLUSION

Based on the research results, the DNN model is effective in predicting heart disease with AdaGrad optimization which provides different performance. The AdaGrad DNN optimization architecture has an accuracy rate of 91.85% with details: 11 input layers, 4 hidden layers with a number of neurons 32, 64, 128, 256, and 1 output layer. For activation, the Input and Hidden Layers use ReLU while the Output Layer uses Sigmoid. The model was trained for 100 epochs with a batch size of 50 and optimized using AdaGrad with a learning rate of 0.01.

REFERENCES

[1] A. Mehmood *et al.*, "Prediction of Heart Disease Using Deep Convolutional Neural Networks," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3409–3422, 2021, doi: 10.1007/s13369-020-05105-1.

[2] J. A. Ramirez-Bautista, A. Hernández-Zavala, S. L. Chaparro-Cárdenas, and J. A. Huerta-Ruelas, "Review on plantar data analysis for disease diagnosis," *Biocybern. Biomed. Eng.*, vol. 38, no. 2, pp. 342–361, 2018.

[3] P. Balakumar, K. Maung-U, and G. Jagadeesh, "Prevalence and prevention of cardiovascular disease and diabetes mellitus," *Pharmacol. Res.*, vol. 113, pp. 600–609, 2016.

[4] N. K. Kumar, G. S. Sindhu, D. K. Prashanthi, and A. S. Sulthana, "Analysis and prediction of cardio vascular disease using machine learning classifiers," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 15–21.

[5] S. Islam, N. Jahan, and M. E. Khatun, "Cardiovascular disease forecast using machine learning paradigms," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 487–490.

[6] D. Zhang *et al.*, "Heart Disease Prediction Based on the Embedded Feature Selection Method and Deep Neural Network," *J. Healthc. Eng.*, vol. 2021, no. Ml, 2021, doi: 10.1155/2021/6260022.

[7] B. Dun, E. Wang, and S. Majumder, "Heart disease diagnosis on medical data using ensemble learning," *Comput. Sci*, vol. 1, pp. 1–5, 2016.

[8] G. Guidi, M. C. Pettenati, P. Melillo, and E. Iadanza, "A machine learning system to improve heart failure patient assistance," *IEEE J. Biomed. Heal. informatics*, vol. 18, no. 6, pp. 1750–1756, 2014.

[9] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Using misclassification analysis for data cleaning," in *International Workshop on Advanced Computational Intelligence and Intelligent Informatics, IWACIII 2009*, 2009.

[10] M. Seyednourani, "A robust process model with two-stage optimization methodology for liquid composite molding process," 2020.

[11] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J. Big Data*, vol. 7, no. 1, pp. 1–41, 2020.

[12] D. K. Thara, B. G. PremaSudha, and F. Xiong, "Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques," *Pattern Recognit. Lett.*, vol. 128, pp. 544–550, 2019.

[13] W. S. E. Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

[14] I. Alfarobi, T. A. Tutupoly, and A. Suryanto, "Komparasi Algoritma C4.5, Naive Bayes Dan Random Forest Untuk Klasifikasi Data Kelulusan Mahasiswa Jakarta," *Mitra dan Teknol. Pendidik*, vol. IV, no. 1, pp. 1–14, 2018.

[15] I. Mackie, *Introduction to Deep Learning*. Springer, 2018. doi: 10.1007/978-3-319-73004-2.

[16] T. Djatna, M. K. D. Hardhienata, and A. F. N. Masruriyah, "An Intuitionistic Fuzzy Diagnosis Analytics for Stroke Disease," *J. Big Data*, pp. 1–14, 2018, doi: 10.1186/s40537-018-0142-7.

[17] M. N. Nasir and I. Budiman, "Perbandingan Pengaruh Nilai Centroid Awal Pada Algoritma K-Means Dan K-Means ++ Confusion Matrix," *Semin. Nas. Ilmu Komput.*, vol. 1, pp. 118–127, 2017.