

Customer Decision Prediction Using Deep Neural Network on Telco Customer Churn Data

Hiya Nalatissifa ^{a,*}, Hilman Ferdinandus Pardede ^{a,b}

^a Graduate School for Computer Sciences
Universitas Nusa Mandiri
Jl. Raya Margonda No.545, Pondok Cina, Beji
Depok, Indonesia

^b Research Center for Informatics
Indonesian Institute of Sciences
Jl. Cisarua No. 21/154D
Bandung, Indonesia

Abstract

Customer churn is the most important problem in the business world, especially in the telecommunications industry, because it greatly influences company profits. Getting new customers for a company is much more difficult and expensive than retaining existing customers. Machine learning, part of data mining, is a sub-field of artificial intelligence widely used to make predictions, including predicting customer churn. Deep neural network (DNN) has been used for churn prediction, but selecting hyperparameters in modeling requires more time and effort, making the process more challenging for the researcher. Therefore, the purpose of this study is to propose a better architecture for the DNN algorithm by using a hard tuner to obtain more optimal hyperparameters. The tuning hyperparameter used is random search in determining the number of nodes in each hidden layer, dropout, and learning rate. In addition, this study also uses three variations of the number of hidden layers, two variations of the activation function, namely rectified linear unit (ReLU) and Sigmoid, then uses five variations of the optimizer (stochastic gradient descent (SGD), adaptive moment estimation (Adam), adaptive gradient algorithm (Adagrad), Adadelta, and root mean square propagation (RMSprop)). Experiments show that the DNN algorithm using hyperparameter tuning random search produces a performance value of 83.09 % accuracy using three hidden layers, the number of nodes in each hidden layer is [20, 35, 15], using the RMSprop optimizer, dropout 0.1, the learning rate is 0.01, with the fastest tuning time of 21 seconds. Better than modeling using k-nearest neighbor (K-NN), random forest (RF), and decision tree (DT) as comparison algorithms.

Keywords: customer churn, data mining, machine learning, deep neural network.

I. INTRODUCTION

Customer churn is the percentage of customers who have stopped or switched to using the product/service periodically [1]. The existence of various service providers allows customers to choose the service provider they want and have the right to switch from one service provider to another [2]. Customer churn is the most crucial problem in the business world because it significantly influences company profits, especially in the telecommunications industry [3].

Currently, the telecommunications industry is experiencing significant problems regarding customer switching due to intense competition and the existence of new, more attractive offers [4]. The telecommunications industry experiences an average annual churn rate of 30 – 35 %, and acquiring new customers is 5 - 10 times more expensive than retaining an existing one [5]. Therefore, based on the observations of a telecommunications company, getting new customers is much more complex and more expensive than retaining existing customers [6].

Churn prediction can be used to help companies identify churners early before a customer shift occurs [7]. Churn prediction can also help companies retain customers and can help companies determine the right marketing strategy to increase profits for the company [8]. In predicting churn, it takes a technique to manage data. To find valuable information from data, the data mining process is applied [9]. Machine learning, part of data mining, is a sub-field of artificial intelligence that is widely researched and used in making predictions, including predictions of customer churn [10].

Research related to utilizing machine learning techniques regarding predictive analysis of customer churn in telecommunications services was carried out by [11] using a comparison of 3 machine learning algorithms, namely deep neural network (DNN), extreme gradient boosting (XGBoost), and random forest (RF). This study shows that the DNN algorithm gives the best results. In this study, the DNN model only focuses on the use of four layers and the dropout on each hidden layer. While the research that the author will do is test the hyperparameter variations in the DNN model, including testing the number of hidden layers, variations of the optimizer used, and variations in learning rates.

Research conducted by [3] proposes a predictive approach to large telecommunications data that can be used to develop models that can predict, classify, and

* Corresponding Author.
Email: 14002363@nusamandiri.ac.id
Received: August 02, 2021 ; Revised: September 27, 2021
Accepted: December 20, 2021 ; Published: December 31, 2021

explain customer churn problems. This study compares several traditional methods. Although the results show that the Random Forest algorithm gets the highest score, it obtained a higher accuracy in the previous year's research using the DNN method. Churn prediction in the banking sector by testing several hyperparameters on the DNN algorithm was carried out in [12]. The test hyperparameters include optimizer variation, activation function, and batch size. This study shows that the DNN algorithm can produce significant predictions in churn prediction by using the rectifier configuration in the hidden layer and the sigmoid activation function in the output layer. Despite the promising results, this study reveals that a longitudinal study is needed to test the productivity of experiments using more samples of data collected over a long period from different banks. Research using exploratory data analysis (EDA) on telco customer churn data was carried out by [7]. This study compares seven machine learning algorithms, namely naïve Bayes (NB), generalized linear model (GLM), logistic regression (LR), deep learning (DL), decision tree (DT), RF, dan gradient boosted tree (GBT). The results showed that all classifications had more than 70 % accuracy and the GBT algorithm had superior accuracy compared to other algorithms. This research still requires optimization of the method used to produce higher accuracy.

Research conducted by [13] compared the results of testing on customer churn data in the telecommunications industry using the NB algorithm with sequential feature selection techniques including sequential forward selection (SFS), sequential backward selection (SBS), sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS). Although the feature selection technique was carried out, the accuracy results were still low in this study. Research on customer churn prediction on telecommunication data was also carried out by [14] using a multi-layered artificial neural network (ANN) algorithm. This research only focuses on one hyperparameter, namely using the Nadam optimizer and five layers. Then research using the comparison of the k-nearest neighbor (K-NN), RF, and XGBoost algorithms to predict customer churn in telecommunications companies was carried out by [15]. In this study, it is necessary to compare other methods to provide more optimal results.

Research with different datasets regarding customer churn was carried out by [16] by selecting features and extraction techniques to obtain efficient features and provide accurate prediction results. The results showed that this technique could produce higher accuracy in the RF algorithm than the original prediction model. [2] conducted research related to telco customer churn prediction using temporal feature engineering, which was then carried out by the ensemble process using RF, XGBoost, and GBT + support vector machine (SVM) algorithms. The results of the study reveal that the use of these features can improve prediction performance. Research on customer churn prediction in the telecommunications industry sector using different datasets was also carried out by [17] using a comparison of classification techniques and grouping techniques to

identify customer churn. This study uses a feature selection technique using information gain and correlation attributes. The best result in this study was obtained by the RF algorithm. In comparison, the research conducted by [18] utilizes the LR model in predicting customer churn in the telecommunications industry. This study performs EDA using visualization and statistical tests for feature selection.

Currently, the DNN algorithm has been used for churn prediction, but selecting hyperparameters in modeling requires more time and effort, which makes the process more challenging for researchers [12]. Therefore, the purpose of this study is to propose a DNN algorithm architecture by tuning hyperparameters to get more optimal results. The hyperparameter tuning or hyperparameter tuning used is random search because, in terms of tuning time, random search spends less time than Bayesian optimization and grid search [19]. The K-NN, RF, and DT methods are also used as other methods that are used as comparison materials to get a model with better performance.

II. PROPOSED METHOD

A. Deep Neural Network Model

DNN is one of the methods that started the emergence and success of Deep Learning [20]. DNN is a neural network-based algorithm that can be used in decision-making. The purpose of DNN is to imitate the workings of the human brain with the Multilayer method. This DNN consists of several hidden layers with connections between layers but no connection between units in each layer. This approach allows complex data to be more easily modeled [21].

DNN has more than three layers (input, hidden, and output layer). In other words, Multilayer perceptron with more layers. Because the layers are relatively many, it is called deep. The learning process at DNN is known as Deep Learning [22]. There is no definite rule of thumb to select the optimal number of hidden layers and their nodes. However, an analytical rule is most often relied upon, which is the optimal size of the hidden layer, usually between the input and output layers [23].

The DNN architecture in this study uses three variations of the number of hidden layers and two variations of the activation function. Each hidden layer uses the rectified linear unit (ReLU) activation function, and in contrast, the output layer uses the Sigmoid activation function. During the training process, manually add the optimizer. Meanwhile, hyperparameter tuning random search determines the number of nodes in each hidden layer, dropout, and learning rate. The architecture of the deep neural network is portrayed in Figure 1.

1) Layer

The three layers of the DNN consist of the input layer, which is the input data layer, the hidden layer receives data from the input layer to be processed in the DNN. In the output layer, the output nodes are binary, resulting in classification decisions [24]. The first variation in this study uses three hidden layers because adding a hidden layer can increase network capacity.

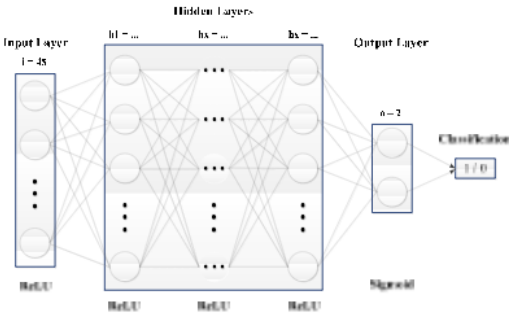


Figure 1. Deep Neural Network (DNN) Architecture.

That is, the more hidden layers will extend the learning stage so that the resulting network is able to map inputs and targets more precisely. However, if the modeling has a too large capacity, it will sometimes cause overfitting [25].

In each hidden layer variation, the number of nodes in the input layer is the same as the input data dimension, which is 45, while the number of nodes in each hidden layer usually has a value smaller than the input data dimension and the number of nodes in the next hidden layer is half of the number of nodes in the hidden layer previously [23]. In comparison, the number of nodes in the output layer is the number of classes contained in the dataset, namely 2 (churn and non-churn).

2) Dropout

It is a process of preventing overfitting and also accelerating the learning process. For this reason, it is necessary to use hyperparameter tuning in determining dropouts. Dropout refers to removing neurons in the form of hidden and visible layers in the network. By removing a neuron, means removing it temporarily from the existing network. The neurons to be removed will be randomly selected. Each neuron will be assigned a probability p between 0 and 1.0 [26].

3) Learning Rate

Learning rate is a training parameter to calculate the weight correction value during the training process. The range of constant values (α) is in the range of zero (0) to one (1). The greater the learning rate, the more likely it will reach the global minimum, but this can also lead to no convergence or not achieving the desired optimal solution. Then, the smaller the learning rate, will reach the minimum and converge, but the learning rate that is too small will cause the process of achieving the optimal solution to be prolonged [27]. Therefore, the application of hyperparameter tuning to get the correct learning rate needs to be done.

4) Activation Function

The activation function is a function that will transform an input into a certain output [28]. And so that the output of the neurons is in the form of a more measurable size, in this study using variations of the ReLu and Sigmoid activation functions.

5) Optimizer

Furthermore, this study tested the optimizer variation [29] to get a more adaptive optimization.

Optimizer is an algorithm used for gradient optimization on neural networks such as weights and learning rates to reduce losses [30]. This study uses five optimizations derived from the Keras model, namely the optimizer stochastic gradient descent (SGD) is a deep learning model that optimizes functions by following a gradient that has noisy with decreasing step size [31], optimization algorithm adaptive moment estimation (Adam) which is an extension for SGD [32], adaptive gradient algorithm (Adagrad) is a modified stochastic gradient descent algorithm with a learning rate per parameter [30], Adadelta is a stochastic gradient descent method based on the dimension adaptive learning rate, and root mean square propagation (RMSProp) is a method in which the learning rate is adjusted for each parameter.

The five optimizers used in this study have been used in various deep learning tasks, and they have achieved excellent results based on these five optimizers [29].

B. Hyperparameter Tuning Random Search

The purpose of hyperparameter tuning is to get optimal hyperparameters [33]. The tuning hyperparameters in this study include the number of hidden units, dropouts, and learning rates.

Hyperparameter random search tries all combinations of hyperparameters in the hyperparameter space. The number of hyperparameter combinations attempted by random search is explicitly specified. Hyperparameter tuning random search is performed using the RandomSearch function of the loud tuner.

The DNN hyperparameters that will be applied in the hyperparameter tuning process are shown in Table 1.

In the modeling process, the analysis process on each data variable is carried out so that the data can be used. Data analysis is carried out to reduce the complexity of unimportant data, detect or remove irrelevant elements and avoid noise from the data [34]. In this study, data preprocessing was carried out with data cleaning and data transforming.

1) Data Cleaning

Checking and deleting data related to missing data values, the aim is to avoid data anomalies at the next stage. In this dataset, there are 11 missing data values, namely the total change attribute.

2) Data Transformation

This stage is done by encoding the nominal data one-hot encoding. The transformation process on the dataset used is using the pandas' library with the get dummies method.

III. EXPERIMENTAL SETUP

The experimental setup in this research includes dataset, modeling, and evaluation model.

TABLE 1
HYPERPARAMETER SEARCH SPACE

Hyperparameters	Values
Number of hidden units	Range 5 to 45
Dropout rate	[0.0; 0.1; 0.2; 0.3]
Learning Rate	[0.1; 0.01; 0.001]

A. Dataset

This study uses a collection of IBM telecommunications data downloaded via Kaggle with the URL address (<https://www.kaggle.com/blatchar/telco-customer-churn>). The data obtained in this study amounted to 7043 data with 21 attributes that will be processed to produce customer churn predictions at telecommunications companies. The attributes of the dataset are described in Table 2.

In the dataset used in this study, there are 5174 or 73.4 % of non-churn customer data, and there are 1869 or 26.6 % churn customer data. Furthermore, in this study, 90 % of the data is used as training data, and 10 % of the data is used as testing data. The churn histogram is portrayed in Figure 2.

B. Modeling

The dataset modeling process in this study carried out four models, namely using the DNN algorithm by testing hyperparameter tuning and using the K-NN algorithm, RF, and DT as other methods that are used as a comparison material to get a model with better performance. The modeling process in this study uses the Python Library version 3 with Google Colab as the tool.

C. Evaluation Model

The research at the evaluation stage compares the accuracy of the results achieved from the four modeling algorithms used, namely the accuracy results from the DNN algorithm by testing using hyperparameter tuning

TABLE 2
ATTRIBUTE NAMES AND DATA TYPES

No.	Attribute Name	Data Type
1	Customer ID	Object
2	Gender	Object
3	Senior Citizen	Int64
4	Partner	Object
5	Dependents	Object
6	Tenure	Int64
7	Phone Service	Object
8	Multiple Lines	Object
9	Internet Service	Object
10	Online Security	Object
11	Online Backup	Object
12	Device Protection	Object
13	Tech Support	Object
14	Streaming TV	Object
15	Streaming Movies	Object
16	Contract	Object
17	Paperless Billing	Object
18	Payment Method	Object
19	Monthly Changes	Float64
20	Total Charges	Object
21	Churn	Object

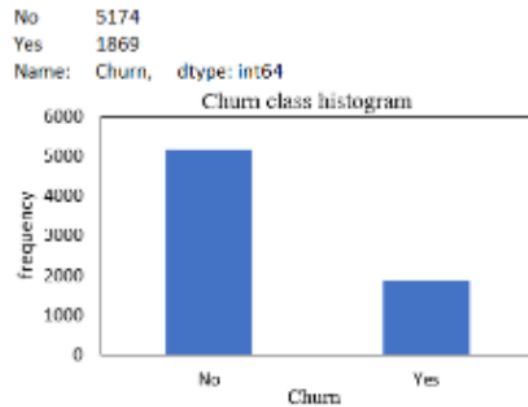


Figure 2. Churn Histogram.

random search and the accuracy results from the K-NN, RF, and DT.

IV. RESULTS AND DISCUSSION

In DNN modeling, hyperparameter tuning random search testing is performed to determine the number of nodes in each hidden layer, dropout, and learning rate.

Then, testing was carried out using three variations of the number of hidden layers, two variations of the activation function, namely ReLu and Sigmoid, and five variations of the optimizer (SGD, Adam, Adagrad, Adadelata, and RMSprop). This model was trained in 50 epochs and 30 batch sizes.

The results of optimizing the hyperparameter DNN and setting the hyperparameter random search time for two hidden layers are shown in Figure 3.

Based on Table 3, the results of the hyperparameter tuning test on two hidden layers and all optimizers obtained the optimal hyperparameter number of nodes in the first hidden layer is 20, and the number of nodes in the second hidden layer is 35. While the optimal hyperparameter for dropout is 0.1 and the learning rate is 0.1. The highest accuracy result in testing two hidden

TABLE 3
HYPERPARAMETER TUNING 2 HIDDEN LAYER RESULT

Optimizer	Optimal Hyperparameters			Tuning Time	Acc.
	Node HL	DO	LR		
SGD	[20, 35]	0.1	0.1	21 s	82.38 %
Adam	[20, 35]	0.1	0.1	41 s	82.38 %
Adagrade	[20, 35]	0.1	0.1	21 s	82.10 %
Adadelata	[20, 35]	0.1	0.1	21 s	81.81 %
RMSprop	[20, 35]	0.1	0.1	41 s	81.67 %

```
Results summary
Results in KerasTuner/Churn_Cust_SGD_Epochs_50_batch_30
Showing 10 best trials
Objective(name='val_accuracy', direction='max')
Trial summary
Hyperparameters:
num_layers: 2
units_0: 20
units_1: 35
dropout_rate: 0.1
learning_rate: 0.1
Score: 0.8238636255264282
```

Figure 3. Hyperparameter Tuning Results for 2 Hidden Layers.

layers of 82.38 % was obtained by the SGD optimizer with a tuning time of 21 seconds.

While the results of optimizing the hyperparameter deep neural network (DNN) and setting the hyperparameter random search time for the three hidden layers are as seen in Table 4.

Based on Table 4, the results of the hyperparameter tuning test on three hidden layers and all optimizers obtained the optimal hyperparameter the number of nodes in the first hidden layer is 20, the number of nodes in the second hidden layer is 35, and the number of nodes in the third hidden layer is 15. In comparison, the hyperparameter optimal for dropout is 0.1 and 0.01 learning rate. The highest accuracy result in the three hidden layer tests of 83.09 % was obtained by the RMSprop optimizer with a tuning time of 21 seconds.

Moreover, optimizing the hyperparameter DNN and setting the hyperparameter random search time on the four hidden layers is shown in Table 5.

Table 5 shows the results of the hyperparameter tuning test on four hidden layers on all optimizers. The optimal hyperparameter obtained is the number of nodes in the first hidden layer is 20, the number of nodes in the second hidden layer is 35, the number of nodes in the third hidden layer is 15, and the number of nodes on the fourth hidden layer is 10. Meanwhile, the optimal hyperparameter for dropout is 0.3, and the learning rate is 0.1. The highest accuracy result in testing four hidden layers of 82.67 % was obtained by the Adagrade optimizer with a tuning time of 24 seconds.

Based on the results of tests that have been carried out on the DNN algorithm by applying hyperparameter tuning, the highest results are obtained in three hidden layers with the order of the number of nodes in each hidden layer, namely [20, 35, 15], using the RMSprop optimizer, dropout 0.1, learning rate 0.01, with the fastest tuning time of 21 seconds. The second highest accuracy result is obtained by four hidden layers with an accuracy value of 82.67 %, with the order of the number of nodes

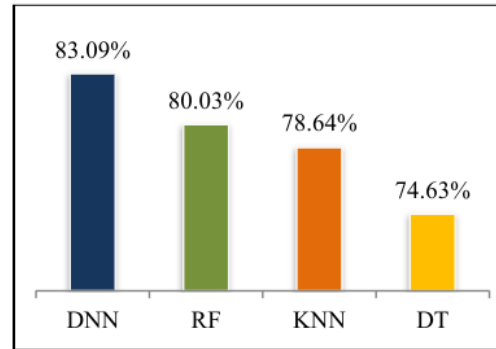


Figure 4. Comparison of Machine Learning Algorithm Accuracy Results.

in each hidden layer, namely [20, 35, 15, 10], using the Adagrade optimizer, dropout 0.3, learning rate 0.1, with a tuning time of 24 seconds. And the last highest accuracy results were obtained by two hidden layers with an accuracy value of 82.38 %, with the order of the number of nodes in each hidden layer, namely [20, 35], using the SGD optimizer, dropout 0.1, learning rate 0.1, with a tuning time of 21 seconds.

The results of tests that have been carried out using hyperparameter tuning on hidden layer variations show that the larger the hidden layer, the greater the dropout required and the longer the processing time required. Furthermore, the test results of several machine learning algorithms used in this study are shown in Figure 4.

Based on the test results in Figure 4, the accuracy gain of the highest value in the machine learning algorithm is 83.09 %, the DNN algorithm is obtained, 80.03 % is the RF algorithm, then the accuracy value is 78.64 % is obtained by the K-NN algorithm, and the lowest accuracy is obtained with a value of 74.63 % in the DT algorithm.

CONCLUSION

This study aims to create a model for predicting customer churn in the telecommunications industry. The dataset used is secondary data downloaded via Kaggle. In this study, the modeling used is a DNN. The selection and use of appropriate hyperparameters using three hidden layers in the DNN modeling in this study resulted in better accuracy performance than the accuracy performance in KNN modeling, RF, and DT. The results of research experiments using hyperparameter tuning random search resulted in DNN modeling with an accuracy value of 83.09 % performance using three hidden layers, the number of nodes in each hidden layer is [20, 35, 15], using the RMSprop optimizer, dropout 0.1, learning rate 0.01, with the fastest tuning time of 21 seconds.

REFERENCES

- [1] H. N. Irmanda, R. Astriratma, and S. Afrizal, "Perbandingan metode jaringan syaraf tiruan dan pohon keputusan untuk prediksi churn," *J. Sist. Inf.*, vol. 11, no. 2, pp. 1817–1825, 2019, doi: 10.36706/jsi.v11i2.9286.
- [2] R. Bai et al., "Context aware telco churn prediction powered by temporal feature engineering," 2018 IEEE Int. Conf. Pervasive Comput. Commun. Workshop, 2018, pp. 161–166, doi: 10.1109/PERCOMW.2018.8480416.

TABLE 4
HYPERPARAMETER TUNING 3 HIDDEN LAYER RESULTS

Optimizer	Optimal Hyperparameters			Tuning Time	Acc.
	Node HL	DO	LR		
SGD	[20, 35, 15]	0.1	0.01	23 s	82.10 %
Adam	[20, 35, 15]	0.1	0.01	42 s	82.81 %
Adagrade	[20, 35, 15]	0.1	0.01	23 s	82.24 %
Adadelta	[20, 35, 15]	0.1	0.01	24 s	79.97 %
RMSprop	[20, 35, 15]	0.1	0.01	21 s	83.09 %

TABLE 5
HYPERPARAMETER TUNING 4 HIDDEN LAYER RESULT

Optimizer	Optimal Hyperparameters			Tuning Time	Acc
	Node HL	DO	LR		
SGD	[20, 35, 15, 10]	0.3	0.1	41 s	82.67 %
Adam	[20, 35, 15, 10]	0.3	0.1	42 s	73.57 %
Adagrade	[20, 35, 15, 10]	0.3	0.1	24 s	82.67 %
Adadelta	[20, 35, 15, 10]	0.3	0.1	21 s	81.96 %
RMSprop	[20, 35, 15, 10]	0.3	0.1	26 s	73.57 %

- [3] D. R. Chabumba, R. Ajoodha, and A. Jadhav, "Predicting telecommunication customer churn using machine learning techniques," in *Proc. Int. Conf. Interdisciplinary Research in Technology and Management*, Feb. 2021.
- [4] T. Utami, D. A. Shofiana, and Y. Heningtyas, "Penerapan algoritma c4 . 5 untuk prediksi churn rate pengguna jasa telekomunikasi," vol. 8, no. 2, pp. 69–76, 2020.
- [5] Y. T. Utami, D. A. Shofiana, and Y. Heningtyas, "Penerapan algoritma C4.5 untuk prediksi churn rate pengguna jasa telekomunikasi," *J. Komputasi*, vol. 8, no. 2, pp. 69–76, 2020, doi: 10.23960%2Fkomputasi.v8i2.2647.
- [6] I. M. M. Mitkees, S. M. Badr, A. I. B. ElSeddawy, "Customer churn prediction model using data mining techniques," *2017 13th Int. Computer Engineering Conf.*, 2017, pp. 262–268, DOI: 10.1109/ICENCO.2017.8289798.
- [7] A. S. Halibas, A. C. Matthew, I. G. Pillai, J. H. Reazol, E. G. Delvo, and L. B. Reazol, "Determining the intervening effects of exploratory data analysis and feature engineering in telecoms customer churn modeling," *2019 4th Middle East College Int. Conf. Big Data Smart City*, 2019, doi: 10.1109/ICBDSC.2019.8645578.
- [8] N. W. Wardani *et al.*, "Prediksi customer churn dengan algoritma decision tree C4.5 berdasarkan segmentasi pelanggan untuk mempertahankan pelanggan pada perusahaan retail," *Rekayasa Sist. Komput.*, vol. 1, no. 1, pp. 16–24, 2018, doi: 10.31598/jurnalresistor.v1i1.219.
- [9] K. Kim and J. Lee, "Bayesian optimization of customer churn predictive model," *2018 Joint 10th Int. Conf. Soft Comput. Intell. Syst. and 19th Int. Symp. Adv. Intell. Syst.*, 2018, pp. 85–88, doi: 10.1109/SCIS-ISIS.2018.00024.
- [10] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan machine learning dalam berbagai bidang: Review paper," *Indones. J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.
- [11] N. P. H. Oka and A. S. Arifin, "Telecommunication service subscriber churn likelihood prediction analysis using diverse machine learning model," *2020 3rd Int. Conf. Mechanical, Electronics, Computer, and Industrial Technology*, 2020, pp. 24–29, doi: 10.1109/MECnIT48290.2020.9166584.
- [12] E. Domingos, B. Ojeme, and O. Daramola, "Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector," *Computation*, vol. 9, no. 3, 2021, Art. no. 34, doi: 10.3390/computation9030034.
- [13] Y. Yulianti and A. Saifudin, "Sequential feature selection in customer churn prediction based on naive Bayes sequential feature selection in customer churn prediction based on naive Bayes," *Institute of Physics Conf. Series: Materials Science and Engineering*, vol. 879, 2020, Art. no. 012090, doi: 10.1088/1757-899X/879/1/012090.
- [14] S. Agrawal, A. Das, A. Gaikwad, and S. Dhage "Customer churn prediction modeling based on behavioural patterns analysis using deep learning," *2018 Int. Conf. Smart Comput. Electron. Enterp.*, 2018, doi: 10.1109/ICSCCE.2018.8538420.
- [15] J. Pamina *et al.*, "An effective classifier for predicting churn in telecommunication," *J. Adv. Res. in Dynamical and Control Syst.*, vol. 11, pp. 221–229, 2019.
- [16] E. Varun *et al.*, "An efficient technique for feature selection to predict customer churn in telecom industry," *2019 1st Int. Conf. Adv. Inf. Technol.*, 2019, pp. 174–179, doi: 10.1109/ICAIT47043.2019.8987317.
- [17] I. Ullah *et al.*, "A churn prediction model using random forest : Analysis of machine learning techniques for churn prediction and factor identification in telecom sector," *IEEE Access*, vol. 7, pp. 60134–60149, 2019, doi: 10.1109/ACCESS.2019.2914999.
- [18] B. N. K. Sai, and T. Sasikala, "Predictive analysis and modeling of customer churn in telecom using machine learning technique," in *Proc. 2019 3rd Int. Conf. Trends Electron. Informatics*, 2019, pp. 6–11, doi: 10.1109/ICOEI.2019.8862625.
- [19] F. F. Firdaus, H. A. Nugroho, and I. Soesanti "Deep neural network with hyperparameter tuning for detection of heart disease," *2021 IEEE Asia Pacific Conf. Wireless and Mobile*, 2021, pp. 59–65, doi: 10.1109/APWiMob51111.2021.9435250.
- [20] E. Rasywir, R. Sinaga, and Y. Pratama, "Evaluasi pembangunan sistem pakar penyakit tanaman sawit dengan metode deep neural network (DNN)," *J. Media Inf.*, vol. 4, no. 4, pp. 1206–1215, 2020, doi: 10.30865/mib.v4i4.2518.
- [21] M. R. Adi, A. B. Osmond, and A. L. Prasasti, "Penentuan dialek jawa menggunakan metode deep neural network," *eProc. Eng.*, vol. 6, no. 2, 2019, pp. 5637–5647.
- [22] B. W. Putra *et al.*, "Klasifikasi aritmia pada sinyal EKG menggunakan deep neural network," *J. Penelitian Ilmu dan Teknologi Komputer*, vol. 13, no. 1, pp. 29–38, 2021.
- [23] W. Treethidaphat, W. Pattara-atikom, and S. Khaimook, "Bus arrival time prediction at any distance of bus route using deep neural network model," *2017 IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 988–992, doi: 10.1109/ITSC.2017.8317891.
- [24] M. Astiningrum, M. Mentari, and R. R. N. Rachma, "Deteksi kesegaran daging sapi berdasarkan ekstraksi fitur warna dan tekstur," in *Proc. 2019: Seminar Informatika Aplikatif*, 2019, pp. 217–222.
- [25] M. Syam, J. Raharjo, and R. Patmasari, "Identifikasi asal daerah berdasarkan suara manusia dengan metode linier predictive coding (lpc)," *eProc. Eng.*, vol. 6, no. 3, pp. 10226–10233, 2019.
- [26] H. Abhirawa, Jondri, and A. Arifianto, "Pengenalan wajah menggunakan convolutional neural network," *eProc. Eng.*, vol. 4, no. 3, pp. 4907–4916, 2017.
- [27] C. D. Suhendra and A. C. Saputra, "Penentuan parameter learning rate selama pembelajaran jaringan syaraf tiruan backpropagation menggunakan algoritma genetika," *J. Tek. Inform.*, vol. 14, no. 2, pp. 202–212, 2020, doi: 10.47111/JTI.
- [28] A. C. Sitepu and M. Sigi, "Analisis fungsi aktivasi relu dan sigmoid menggunakan optimizer SGD dengan representasi MSE pada model backpropagation," *J. Teknik Inform. Universal*, vol. 1, no. 1, pp. 12–25, 2021.
- [29] Y. Wang, J. Liu, J. Mišić, V. B. Mišić, and S. Lv, "Assessing optimizer impact on DNN model sensitivity to adversarial examples," *IEEE Access*, vol. 7, pp. 152766–152776, 2019, doi: 10.1109/ACCESS.2019.2948658.
- [30] N. A. Putro, R. Septian, Widiastuti, M. Maulidah, 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, 2020, doi: 10.33480/techno.v18i1.2056.
- [31] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *J. Machine Learning Res.*, vol. 18, pp. 1–35, 2017.
- [32] M. W. P. Aldi, Jondri, and A. Aditsania, "Analisis dan implementasi long short term memory neural network untuk prediksi harga bitcoin," *eProc. Eng.*, vol. 5, no. 2, pp. 3548–3555, 2018.
- [33] M. P. Ranjit, G. Ganapathy, K. Sridhar, and V. Arumugham "Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with bayesian optimization in the cloud," *2019 IEEE 12th Int. Conf. Cloud Comput.*, 2019, pp. 520–522, doi: 10.1109/CLOUD.2019.00097.
- [34] S. Sautomo and H. F. Pardede, "Prediksi belanja pemerintah Indonesia menggunakan long short-term memory (LSTM)," *J. Rekayasa Sist. dan Teknol. Inf.*, vol. 5, no. 1, pp. 99–106, 2021, doi: 10.29207/resti.v5i1.2815.