

Naive Bayes modification for intrusion detection system classification with zero probability

Yogiek Indra Kurniawan¹, Fakhrrur Razi², Nofiyati³, Bangun Wijayanto⁴, Muhammad Luthfi Hidayat⁵

^{1,3,4}Informatics Department, Engineering Faculty, Universitas Jenderal Soedirman, Indonesia

²Informatics Department, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia

⁵Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia

Article Info

Article history:

Received Jan 25, 2021

Revised Apr 30, 2021

Accepted Jul 25, 2021

Keywords:

Algorithm modification

Data mining

Intrusion detection system

Naïve Bayes

Zero probability

ABSTRACT

One of the methods used in detecting the intrusion detection system is by implementing Naïve Bayes algorithm. However, Naïve Bayes has a problem when one of the probabilities is 0, it will cause inaccurate prediction, or even no prediction was found. This paper proposed two modifications for Naïve Bayes algorithm. The first modification eliminated the variable that has 0 probability and the second modification changed the multiplication operations to addition operations. This modification is only applied when the Naïve Bayes algorithm does not find any prediction results caused by zero probabilities. The results of this research show that the value of precision, recall, and accuracy in the modification made tends to increase and better than the original Naïve Bayes algorithm. The highest precision, recall, and accuracy are obtained from modification by changing the multiplication operation to the addition. Increasing precision can reach 4%, increasing recall reaches 2% and increasing accuracy reaches 2%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Yogiek Indra Kurniawan

Informatics Department, Engineering Faculty

Universitas Jenderal Soedirman

Prof. Dr. HR. Boenjamin Street No. 708, Purwokerto, Central Java, Indonesia

Email: yogiek@unsoed.ac.id

1. INTRODUCTION

Network and data security are some of the most important things for an agency at this time. Various types of attacks that occur through the internet against networks and data encourage agencies to implement various systems to detect and prevent attacks that occur [1]. One system that is often used to detect attacks is intrusion detection system (IDS). IDS is a system used to automate the process of detecting suspicious activity in the network and analyze the possibility of attacks in these activities [2], [3]. There are several methods used in IDS to detect, including anomaly detection and misuse detection. Anomaly detection is detection by comparing the state of an existing activity with the state when a normal activity, while misuse detection is detection by matching the activity pattern with a pattern contained in a database that has been previously defined [4]. Apart from these two methods, several studies have been carried out to conduct detection, prediction, or classification using data mining algorithms [5]-[8]. One algorithm that can be used to predict IDS is Naïve Bayes [9], [10] which gives good accuracy.

Naïve Bayes algorithm is a classification algorithm that is quite good and is often used in various studies [11]-[13]. This algorithm can be used for simple classification with fixed Y variable and also for text classification [14]-[16]. Laga and Sarno [17] showed that Naïve Bayes gave the best accuracy from other

classification methods, such as KNN, SVM, and random forest. However, the Naïve Bayes algorithm still has a drawback, that is, if the probability value from one of the variables is 0, it can make the final comparison result 0, which can lead to inaccurate prediction results [15], [17]-[20]. Research [15], [17] overcomes zero probability with RB-Bayes, while research [20] uses Hybrid N-gram, and research [19], [20] uses multinomial Naïve Bayes.

Based on the previous research [15], [17]-[20], it can be seen if the prediction results from the testing data are not found due to the opportunity 0. Therefore, it is necessary to modify the Naïve Bayes algorithm to overcome the existing problems. This paper proposed the modification of Naïve Bayes algorithm to overcome opportunity 0 in the dataset. In this research, the Naïve Bayes algorithm and some Naïve Bayes modifications are implemented in a web-based application and analyze whether the modifications made can improve the accuracy of prediction of attacks in IDS or vice versa. The first modification is to eliminate the variable that has a probability value of 0, while the second modification changes the calculation from multiplication to addition. Both of these modifications are applied when the Naïve Bayes algorithm does not find any classification results.

2. RESEARCH METHOD

The research method used in this paper is in Figure 1. Each stage is carried out in stages and sequentially.

2.1. Problem identification

Problem analysis is the initial stage for identifying a case or problem [21], [22]. This stage is the initial stage which aims to determine the problems that exist in the Naïve Bayes algorithm, especially in predicting attacks in the network. The problem obtained at this stage is that there is an opportunity value of 0 in Naïve Bayes that can make the prediction results inaccurate and the lack of the ability of IDS to predict attacks in the network.



Figure 1. Research method

2.2. Data collection

The data in this study came from the NSL-KDD 99 dataset. NSL-KDD 99 is a dataset resulting from the development and reduction of fundamental problems from the KDD 99 dataset. The dataset used is small training set.csv and KDDTest + .csv [23]. Some of the advantages of the NSL-KDD 99 dataset compared to the original KDD 99 dataset include:

- a. The data contained in the training data is not excessive so the classification results are not biased.
- b. There is no data duplication in the testing data.
- c. The amount of data in training and testing data makes sense, which makes it affordable to run experiments on complete datasets without having to randomly select a small portion.

2.3. Data preprocessing

In this stage, several processes are carried out to process the data before classification is performed using the Naïve Bayes algorithm. The process includes:

- a. data cleansing
- b. feature selection
- c. variable discretization

2.4. Implementation

At this stage, the application starts to be built by the design made in the previous stage. The application is realized in the web form with PHP programming language and using MySQL database.

2.5. Testing

The next stage after implementation is testing the system. This stage is carried out to test the Naive Bayes algorithm and the modifications that have been made. The tests carried out are divided into 2, namely algorithm testing and testing of the precision, recall, and accuracy values.

3. RESULTS AND DISCUSSION

This section is a discussion of the research that has been done. Starting from the preprocessing stage, application implementation, and testing.

3.1. Preprocessing

In this stage, several processes are carried out to process the data before classification is performed using the Naïve Bayes algorithm. This stage is implemented because preprocessing can improve the accuracy of Naive Bayes [24]. The process includes:

3.1.1. Data cleansing

This stage is done to eliminate the data in the testing data with the Y variable that is not contained in the training data and to change the class classification (variable Y) from the previous one as the name of the attack to the type of attack so that the number of Y variables is lower so the system performance can be faster. Attack names and attack types can be seen in Table 1.

Table 1. List of names and types of attacks

Attack Type	Attack Name
Normal	Normal
Dos	Back, Teardrop, Land, Smurf, Pod, Neptune
Probe	Ipsweep, Satan Portsweep Nmap
R2L	guess_passwd, ftp_write, imap, warezmaster, warezclient, multihop, spy, phf
U2R	Rootkit, buffer_overflow, perl, loadmodule

3.1.2. Feature selection

This stage aims to reduce the number of X variables so they are not too many and to improve the accuracy of the predictions produced. The method used in feature selection is correlation-based features selection (CFS). CFS chooses X variables that have the highest correlation with Y variables but has the fewest correlations between X variables. The Feature Selection process in this study was carried out using WEKA tools which produced 10 X variables out of a total of 41 existing X variables. The list of variables are : flag, src_bytes, dst_bytes, hot, logged_in, count, srv_serror_rate, diff_srv_rate, dst_host_diff_srv_rate, and dst_host_srv_diff_host_rate.

3.1.3. Variable discretization

This stage aims to change the variables in the dataset which are of the continuous type to discrete types. The discretization method used in this stage is supervised discretization because the variable X correlates directly with the Y variable. The results of variable discretization are shown in Table 2.

Table 2. Results from variable discretization

X ₂	X ₃	X ₄	X ₅	X ₇	X ₈	X ₉	X ₁₀
0-0.5	0-0.5	0-0.5	0-2.5	0-0.64	0-0.03	0-0.005	0-0.005
0.5-8.5	0.5-2849.5	0.5-inf	2.5-23.5	0.64-inf	0.03-0.145	0.005-0.045	0.005-0.105
8.5-17.5	2849.5-inf		23.5-81.5		0.145-0.355	0.045-0.125	0.105-0.245
17.5-18.5			81.5-313.5		0.355-inf	0.125-inf	0.245-inf
18.5-23.5			313.5-inf				
23.5-28.5							
28.5-333.5							
333.5-334.5							
334.5-518							
518-inf							

3.2. Implementation

At this stage, the application starts to be built by the design made in the previous stage. The application is realized in the web form with PHP programming language and using MySQL database. There are 3 algorithms applied in the application, including:

3.2.1. Naive Bayes

Naïve Bayes is a simple probability classification based on Bayes' Theorem where each feature/variable is assumed to be independent of each other. Bayes' theorem was put forward by a British scientist named Thomas Bayes as a theorem for predicting future opportunities based on experience [24]. The Bayes theorem equation can be seen in (1):

$$P(H|U) = \frac{P(U|H).P(H)}{P(U)} \quad (1)$$

where: U : Data with unknown classes

H : The U data hypothesis is a specific class

$P(H|U)$: The probability of hypothesis H is based on condition U

$P(H)$: Hypothesis probability H

$P(U|H)$: U probability based on conditions

$P(U)$: U Probability

3.2.2. Modification 1

From the example calculation done above, it can be seen if there is a problem where no prediction results are found because all the classes have a probability value of 0. Therefore, in modification 1 this is done by removing variables that have a 0 value, so the probability of each class when there is no comparison 0 value.

3.2.3. Modification 2

In this modification 2, to overcome the probability value of 0 on Naïve Bayes is to change the multiplication operation into an addition so that the probability results of each class are not worth 0.

3.2.4. Application implementation

The implementation of the previously created design resulted in a web-based application to test the modifications made in this study. In this application, there is one admin user who acts as a data manager. Admins are required to log in before they can manage the data in the system. On the main page, several menus can facilitate the admin to manage data, including training data, testing data, and testing page. In the data training and data testing menu, there are submenu namely the view data menu shown in Figure 2.

List Data Training

Show 10 entries

















flag	src_bytes	dst_bytes	hot	logged_in	count	srv_serror_rate	diff_srv_rate	dst_host_diff_srv_rate	dst_host_srv_diff_host_rate	label	Action
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	0 - 2.5	0 - 0.64	0 - 0.03	0 - 0.005	0.105 - 0.245	normal	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	81.5 - 313.5	0 - 0.64	0.03 - 0.145	0.045 - 0.125	0 - 0.005	dos	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	0 - 2.5	0 - 0.64	0 - 0.03	0.005 - 0.045	0 - 0.005	normal	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	81.5 - 313.5	0 - 0.64	0.03 - 0.145	0.045 - 0.125	0 - 0.005	dos	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	0 - 2.5	0 - 0.64	0.355 - 100000000000	0.125 - 100000000000	0 - 0.005	probe	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	81.5 - 313.5	0 - 0.64	0.03 - 0.145	0.045 - 0.125	0 - 0.005	dos	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	81.5 - 313.5	0 - 0.64	0.03 - 0.145	0.045 - 0.125	0 - 0.005	dos	 
REJ	0 - 0.5	0 - 0.5	0 - 0.5	0	0 - 2.5	0 - 0.64	0 - 0.03	0 - 0.005	0.005 - 0.105	normal	 

Figure 2. View training data page

On the manage data page, the admin can input data either through the form provided or through CSV import using the import data button. In addition, the admin can also delete all data that has been entered by using the delete all button.

On the data view page, Admin can view, edit, and delete data that has been entered. On the data testing list, there is a button that can be used to start the classification process. The testing menu can be used to see the results of the classification process that has been carried out by the system. The page views of the tests are shown in Figure 3.

3.3. Testing

Testing is a way to assess quality from an algorithm [25]. This stage is carried out with 2 methods, including algorithm test and precision, recall, and accuracy test.

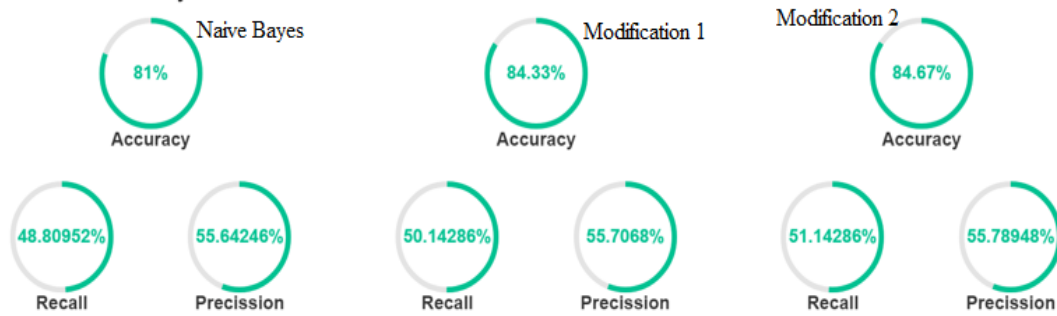


Figure 3. Test page display

3.3.1. Algorithm test

Algorithm testing is done by comparing the results of manual calculations with calculations performed by the application. If the calculation result manually is the same as the calculation result using the application, it indicates that the application has performed the calculation correctly. The comparisons compared are the calculations with the Naive Bayes algorithm, the Naive Bayes algorithm with modification 1, and the Naive Bayes algorithm with the 2nd modification.

This test is carried out using 10 training data and 3 testing data. The calculation is done using manual calculations and calculations with applications that have been built. From the results obtained, manual calculations and calculations using the application have the same results. This shows that the application built has implemented the Naive Bayes algorithm and the two modifications are appropriate.

3.3.2. Precision, recall, and accuracy test

Testing precision, recall, and accuracy is done by calculating the value of precision, accuracy, and recall of the Naive Bayes algorithm and the two modifications made. Precision is a calculation of the estimated proportion of positive cases that is formulated in (2) [26], [27]:

$$precision = \frac{TP}{FP+TP} \tag{2}$$

A recall is a calculation of the estimated proportion of positive cases that are correctly identified and as shown in (3):

$$recall = \frac{TP}{TP+TN} \tag{3}$$

Accuracy is a calculation of the proportion of the total number of correct predictions and as shown in (4):

$$accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{4}$$

where: TP: True Positive
 TN: True Negative
 FP: False Positive
 FN: False Negative

In this test, the testing data used has a fixed amount of 300 data while training data starts from 200 data to 1200 data with the addition of 200 data for each test. This is done to analyze the value of precision, recall, and accuracy of the Naive Bayes algorithm along with the modifications applied. The results of testing precision, recall, and accuracy can be seen in Figures 4, 5, and 6.

3.4. Results analyze

The application that was built in this study has one actor, namely the administrator who has access rights to manipulate training data and data testing and classification testing. Testing of applications that have been built is done by 2 methods, namely algorithm testing and testing precision, recall, and accuracy. The precision, recall, and accuracy testing on the Naive Bayes algorithm and the two modifications showed an increase with increasing training data. This is because increasing the amount of data can increase the possibility of the same data so that increasing the data can increase the precision, recall, and accuracy values

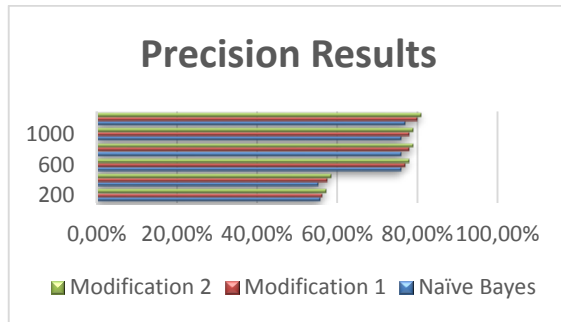


Figure 4. Precision results

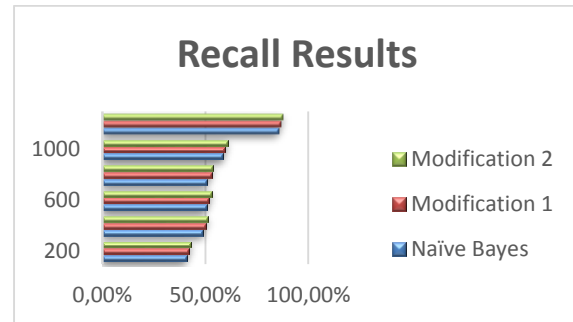


Figure 5. Recall results

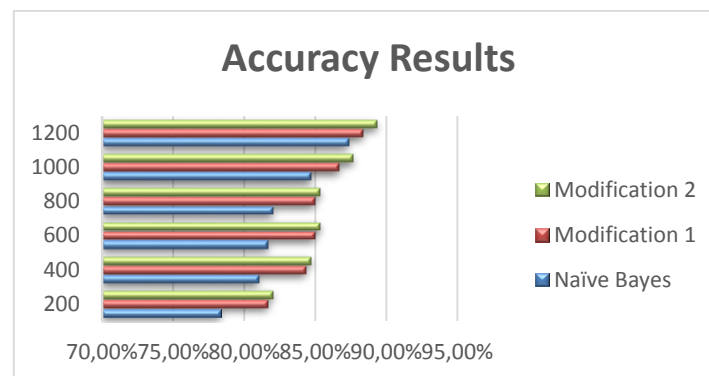


Figure 6. Accuracy results

The maximum value of precision in Naïve Bayes is 76.83% in the training data of 1200 data. While for the same training data, the precision value in modification 1 is 79.83% and the precision value in modification 2 is 80.83%. This shows that there is an increase in precision in modifications 1 and 2 with the highest value obtained by modification 2. The maximum value of recall on Naïve Bayes is 85.52% for 1200 training data. Whereas in the same training data, the recall value in modification 1 was 86.52% and the recall value in modification 2 was 87.52%. This shows that there is an increase in recall on modifications 1 and 2 with the highest value obtained by modification 2. The maximum value of accuracy at Naïve Bayes is 87.33% for 1200 training data. While for the same training data, the accuracy value on modification 1 is 88.33% and the accuracy value on modification 2 is 89.33%. This shows that there is an increase in accuracy at modifications 1 and 2 with the highest value obtained by modification 2. Based on the tests that have been done, it can be concluded that modification by eliminating a variable that has a value of 0 and modification by changing the multiplication operation by addition can increase precision, recall, and accuracy. The highest precision, recall, and accuracy is obtained from modification by changing the multiplication operation with the addition of the value resulting in the possible value of 0. Increasing precision can reach 4%, increasing recall reaches 2% and increasing accuracy reaches 2%.

4. CONCLUSION

Based on the tests that have been done, it can be concluded that the precision, recall, and accuracy testing on the Naïve Bayes algorithm and the two modifications showed an increase with increasing training data. Besides that, modification by eliminating a variable that has a value of 0 and modification by changing the multiplication operation by addition can increase precision, recall, and accuracy. The highest precision, recall, and accuracy is obtained from modification by changing the multiplication operation with the addition of the value resulting in the possible value of 0. Based on the results obtained, to achieve better results it is recommended that improvements be made to the modifications that have been made in subsequent studies.

ACKNOWLEDGEMENTS

We would like to thank “*Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Jenderal Soedirman (UNSOED)*”, Indonesia who has provided funding for this research in the scheme “Riset Dosen Pemula 2020”, as well as to various parties who have helped carry out this research.

REFERENCES

- [1] E. A. Shams and A. Rizaner, “A novel support vector machine based intrusion detection system for mobile ad hoc networks,” *Wirel. Networks*, vol. 24, no. 5, pp. 1821–1829, 2018, doi: 10.1007/s11276-016-1439-0.
- [2] C. Birkinshaw, E. Rouka, and V. G. Vassilakis, “Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks,” *J. Netw. Comput. Appl.*, vol. 136, pp. 71–85, 2019, doi: 10.1016/j.jnca.2019.03.005.
- [3] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” *Ad Hoc Networks*, vol. 90, 2019, doi: 10.1016/j.adhoc.2019.02.001.
- [4] W. Yan and L. Yu, “On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach,” *Proc. Annu. Conf. Progn. Heal. Manag. Soc. PHM*, pp. 440–447, 2015.
- [5] K. Peng, V. C. M. Leung, and Q. Huang, “Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System over Big Data,” *IEEE Access*, vol. 6, pp. 11897–11906, 2018, doi: 10.1109/ACCESS.2018.2810267.
- [6] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse, “Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines,” *Journal of the Franklin Institute*, vol. 355, no. 4, pp. 1752–1779, 2018, doi: 10.1016/j.jfranklin.2017.06.006.
- [7] A. Fadli, A. Wisnu, W. Nugraha, M. S. Aliim, Y. I. Kurniawan, and W. H. Purnomo, “Simple Correlation Between Weather and COVID-19 Pandemic Using Data Mining Algorithms,” in *IOP Conference Series : Materials Science and Engineering 982 012015*, 2020, pp. 1–10, doi: 10.1088/1757-899X/982/1/012015.
- [8] A. F. Z. Abidin *et al.*, “Adaboost-multilayer perceptron to predict the student’s performance in software engineering,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1556–1562, 2019, doi: 10.11591/eei.v8i4.1432.
- [9] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018, doi: 10.1016/j.future.2017.10.016.
- [10] H. A. Mahmood, “Network Intrusion Detection System (NIDS) in Cloud Environment based on Hidden Naïve Bayes Multiclass Classifier,” *Al-Mustansiriyah Journal of Science*, vol. 28, no. 2, p. 134, 2018, doi: 10.23851/mjs.v28i2.508.
- [11] W. Chen, S. Zhang, R. Li, and H. Shahabi, “Science of the Total Environment Performance evaluation of the GIS-based data mining techniques of best- first decision tree , random forest , and naïve Bayes tree for landslide susceptibility modeling,” *Science of The Total Environment*, vol. 644, pp. 1006–1018, 2018, doi: 10.1016/j.scitotenv.2018.06.389.
- [12] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, “Differentially private Naive Bayes learning over multiple data sources,” *Information Sciences.*, vol. 444, pp. 89–104, 2018, doi: 10.1016/j.ins.2018.02.056.
- [13] N. M. Nawi, M. Makhtar, M. Z. Salikon, and Z. A. Afip, “A comparative analysis of classification techniques on predicting flood risk,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1342–1350, 2020, doi: 10.11591/ijeecs.v18.i3.pp1342-1350.
- [14] M. O. Ibrohim and I. Budi, “Translated vs non-translated method for multilingual hate speech identification in Twitter,” *International Journal on Advanced Science Engineering and Information Technology*, vol. 9, no. 4, pp. 1116–1123, 2019, doi: 10.18517/ijaseit.9.4.8123.
- [15] I. R. A. Hamid, S. Subramaniam, E. Sutoyo, and Z. Abdullah, “Classification of polymorphic virus based on integrated features,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 6, pp. 2577–2583, 2018, doi: 10.18517/ijaseit.8.6.5045.
- [16] M. H. Jopri, M. R. Ab Ghaiii, A. R. Abdullah, M. Manap, T. Sutikno, and J. Too, “K-nearest neighbor and naïve bayes based diagnostic analytic of harmonic source identification,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2650–2657, 2020, doi: 10.11591/eei.v9i6.2685.
- [17] S. A. Laga and R. Sarno, “Temperature effect of electronic nose sampling for classifying mixture of beef and pork,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, p. 1626, 2020, doi: 10.11591/ijeecs.v19.i3.pp1626-1634.
- [18] Rajni and Amandeep, “RB-bayes algorithm for the prediction of diabetic in ‘PIMA Indian dataset,’” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, pp. 4866–4872, 2019, doi: 10.11591/ijece.v9i6.pp4866-4872.
- [19] R. Bhalla and A. Bagga, “Opinion mining framework using proposed rb-bayes model for text classification,” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 477–484, 2019, doi: 10.11591/ijece.v9i1.pp477-484.
- [20] J. Awwalu, A. A. Bakar, and M. R. Yaakub, “Hybrid N-gram model using Naïve Bayes for classification of political sentiments on Twitter,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 9207–9220, 2019, doi: 10.1007/s00521-019-04248-z.
- [21] Y. I. Kurniawan, A. Rahmawati, N. Chasanah, and A. Hanifa, “Application for determining the modality preference of student learning,” in *Journal of Physics: Conference Series*, 2019, vol. 1367, no. 1, pp. 1–11, doi: 10.1088/1742-

- 6596/1367/1/012011.
- [22] F. Y. Al Irsyadi, S. Supriyadi, and Y. I. Kurniawan, "Interactive educational animal identification game for primary schoolchildren with intellectual disability," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3058–3064, 2019, doi: 10.30534/ijatcse/2019/64862019.
- [23] C. Low, "NSL-KDD Dataset," 2015, [Online]. Available: https://github.com/defcom17/NSL_KDD (accessed Sep. 13, 2019).
- [24] Y. I. Kurniawan, T. Cahyono, Nofiyati, E. Maryanto, A. Fadli, and N. R. Indraswari, "Preprocessing Using Correlation Based Features Selection on Naive Bayes Classification," in *IOP Conference Series : Materials Science and Engineering* 982 012012, 2020, pp. 1-8, doi: 10.1088/1757-899X/982/1/012012.
- [25] Y. I. Kurniawan, E. Soviana, and I. Yuliana, "Merging Pearson Correlation and TAN-ELR algorithm in recommender system," in *AIP Conference Proceedings*, 2018, vol. 1977, doi: 10.1063/1.5042998.
- [26] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1-9, 2015, doi: 10.1016/j.simpat.2015.03.003.
- [27] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, no. 1, pp. 152–160, 2018, doi: 10.1016/j.jocs.2017.03.006.

BIOGRAPHIES OF AUTHORS



Yogie Indra Kurniawan is a lecturer from Informatics Department, Universitas Jenderal Soedirman, Indonesia. He graduated bachelor's degree from Informatics, Institut Teknologi Telkom, Indonesia, and a master's degree from Informatics, Institut Teknologi Bandung, Indonesia. His research concern is about Data Mining and Software Development.



Fakhur Razi graduated from the Informatics department, Universitas Muhammadiyah Surakarta for a bachelor's degree in 2019. Currently, he is working at PT Security Research Labs with a position as an Ethical Hacker.



Nofiyati is a lecturer from Informatics Department, Universitas Jenderal Soedirman, Indonesia. Nofiyati was born in Kebumen, August 19, 1981. She completed his undergraduate studies in Information Systems at the State Islamic University (UIN) Syarif Hidayatullah Jakarta in 2006 and completed his Masters in Information Systems at Diponegoro University in 2014. Nofiyati has a career as a lecturer since 2007 with the main subject being taught, namely Information Systems Project Management.



Bangun Wijayanto was born in Banyumas, Indonesia. In 2005 received a bachelor's degree from Jenderal Soedirman University. In 2010 received the M.Cs. degrees from Gajah Mada University. His research interest includes Embedded System, Information System and software engineering.



Muhammad Luthfi Hidayat is a Ph.D. student in the Faculty of Computing and Information Technology, Department of Information Technology, at King Abdulaziz University, Saudi Arabia. He is a lecturer and educational-technology researcher at Universitas Muhammadiyah Surakarta, Indonesia.