

Imbalanced data handling in multiclass distributed denial of service attack detection using deep learning

Rahmad Gunawan^{1,2}, Hadhrami Ab Ghani², Nurulaqilla Khamis³, Hasanatul Fu'adah Amran¹

¹Department of Informatics Engineering, Faculty of Computer Sciences, Universitas Muhammadiyah Riau, Pekanbaru, Indonesia

²Department of Data Science, Faculty of Data Science and Computing, Universiti Malaysia Kelantan, Kota Bharu, Malaysia

³Department of Control and Mechatronic, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, Malaysia

Article Info

Article history:

Received Jan 24, 2024

Revised Jun 6, 2024

Accepted Jul 12, 2024

Keywords:

Class-weight

Deep learning

Distributed denial of service

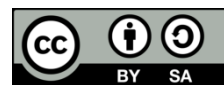
Random oversampling

Random undersampling

ABSTRACT

In data analysis, imbalanced datasets are a frequent issue, where classes in a dataset have an uneven distribution, which can lead to poor performance in machine learning (ML) and predictive modeling. In this study, we analyze distributed denial of service (DDoS) attacks at the application layer. Three primary strategies are studied in this study to address the issue of data imbalance in multiclass techniques: random oversampling (ROS), random undersampling (RUS), and the use of class weights. A model using a deep learning (DL) technique has been proposed in this paper to be trained and tested for DDoS attack detection. Based on the results obtained and presented in this paper, it is observed that RUS outperforms class-weight and ROS in multiclass settings in terms of resolving imbalanced data when implemented with the deep learning-based DDoS attack detection model.

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



Corresponding Author:

Rahmad Gunawan

Department of Informatics Engineering, Faculty of Computer Sciences, University Muhammadiyah Riau
St. Tuanku Tambusai, Pekanbaru, Riau Province, Indonesia

Email: goengoen78@umri.ac.id

1. INTRODUCTION

Distributed denial of service (DDoS) attacks cause concern and losses both in the business world and in government. Denial of service (DOS) attacks have been widely discussed in the world of information technology and have been the subject of discussions and research on internet network attacks since the early 1980s [1]. Since the summer of 1999, when the first spread of (DDoS) assault incidence was recorded, the majority of DoS attacks have been spread in form [2]. Now, according to a 2020 Kaspersky analysis, more than 20% of DOS attacks on businesses worldwide employ a method that leverages several devices to seize control of and harm a particular system or network [3], the incidence of these attacks tripled between the second quarter of 2020 and the corresponding period in 2019. Attacks on the network/transport layer, an entity capable of inundating internet traffic through the generation of an extensive volume of packets is known for its ability to deploy connection packets in a fragmented manner over the user datagram protocol (UDP). Simultaneously, the transmission control protocol (TCP) employs TCP-SYN [4] and rejection techniques, whereas the internet control message protocol (ICMP) and domain name system (DNS) are targeted in the process.

DDoS attacks, in contrast, are malicious attempts characterized by flooding a target system, network, or service with an excessive flow of Internet traffic, thereby disrupting or slowing down its or its service's normal operational functions [5]. Attacks occurring at the application layer are characterized by their minimal bandwidth requirements and are often surreptitious. Application layer assaults are a new kind of DDoS attacks that have started to gain traction recently, this attack takes the use of holes in protocols that function at the application layer [6]. The utilization of deep learning (DL) approaches for addressing DDoS attacks with

imbalanced data at the application layer may lead to an overload of application servers and resource consumption. These attacks are more targeted, aiming to interfere with legitimate user services by attacking applications like DNS, hypertext transfer protocol (HTTP), and lightweight directory access protocol (LDAP) [7]. Although traditional solutions such as regular classifiers have been used for DDoS detection, the main obstacles lie in handling imbalanced data and loss of information in minority classes, which leads to inaccurate predictions and increased false positives in multiclassification. Therefore, this research aims to develop a more effective approach using DL to increase the accuracy of DDoS detection, especially in minority classes, as well as reduce the number of false positives in overall detection.

The most significant problem with all real-world applications is imbalanced data. However, a classifier that is more biased in favor of the majority class is frequently produced when classification accuracy based on the minority class is high. Thus, improving a minority class's classification accuracy is an extremely important task. In the field of data mining, classifying data with imbalanced classes is a significant challenge. Numerous methods have been put forth to address imbalanced data sets; however, they have mostly addressed two-class issues and have given considerably less thought to multiclass scenarios [8], [9]. Protein classification [10], welding faults classification [11], gearbox fault detection [12], pediatric brain tumors [13], categorization of hyperspectral images [14], text categorization [15], and activity identification [16]. This paper focuses on detecting DDoS attacks at the application layer with DL on multiclass by comparing resampling, random undersampling (RUS), random oversampling (ROS), and class-weight techniques.

2. LITERATURE REVIEW

2.1. Multiclass imbalanced classification

This section presents contemporary techniques for identifying unbalanced data sets with more than two classes. Even though several techniques might be included in multiple categories, they are separated into groups [9]. Data level, many writers have taken into consideration data-preprocessing techniques for multiclass imbalanced issues [17], [18].

2.2. Random oversampling

Synthetic minority oversampling technique (SMOTE) and other oversampling techniques are based on the ROS basic approach. Giving researchers access to additional minority class samples enhances the prediction model's overall accuracy and mitigates the issue of class imbalance [19]. As seen in Figure 1, ROS duplicates the minority class's data points at random with replacement until the relative proportions of the two classes are equal.

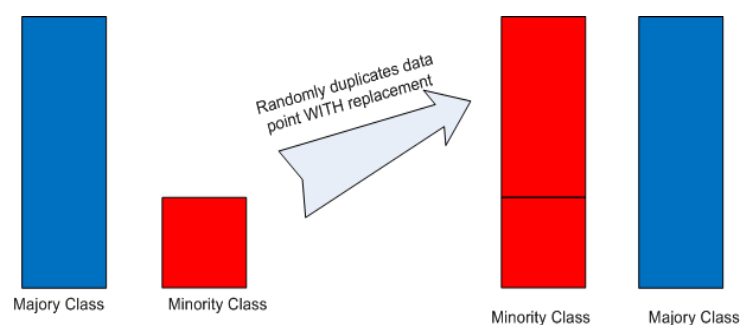


Figure 1. ROS process

RUS, as compared with ROS, works by arbitrarily eliminating the majority class's data points, with or without the replacement, up until the two groups' relative proportions are equal. Using this technique, the targeted variable's instances of the majority class are arbitrarily eliminated until their number equals that of the minority class [20]. The RUS approach applies the following selection process after balancing the majority and minority classes:

$$x_{new} = x_i + (\hat{x}_i - x_i) * \delta, \quad (1)$$

where x_{new} refers to the synthesized instances, x_i refers to the minority class samples, \hat{x}_i is among the neighbors that are k-nearest for x_i , and δ is a representation chosen at random from 0 and 1.

2.3. Random undersampling

The implementation of the undersampling strategy entails reducing the volume of data associated with the majority class, the entirety of the data from the minority class is preserved [20]. The RUS approach is one that's frequently used to address unbalanced datasets by lowering the majority class [21]. Through the utilization of a randomized selection method, many cases from the majority class are eliminated structurally or systematically until a balance is reached in the number of cases between the minority and majority classes. This method's advantage is that it may be applied randomly, negating or lowering the value of the majority class under certain circumstances. The undersampling method's flaw can lead to the removal of significant portions of the majority class data, which can impact classification results.

Figure 2 shows the operation of RUS. Analysts can utilize intricate prediction models, such as the stacked ensemble method, in the course of this project, the application of RUS is directed towards the reduction of the dataset size, with the primary objective of diminishing the training time required for the predictive model. Additionally, RUS contributes to enhancing model efficiency by achieving a more balanced distribution of classes within the relevant variables [22]. However, RUS may exclude significant data and patterns that might aid in the prediction by shrinking the dataset, which would lower the algorithm's forecast accuracy [19], [23].

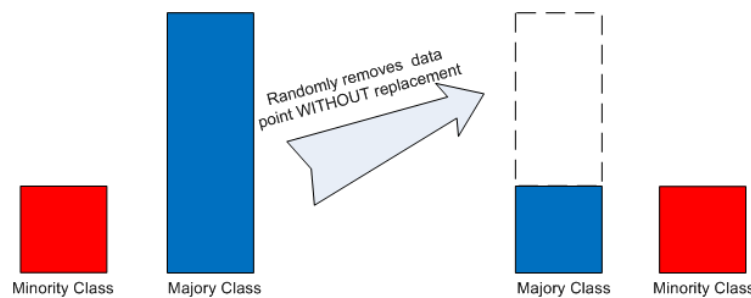


Figure 2. RUS process

2.4. Class-weight

Comparing the weighted class or class-weight method to the other sample techniques, one finds that it takes a different approach [24]. We employed an inverse class frequency variation for our class weight implementation, giving minority classes non-excessive, higher weights. We use (1) and (2) to figure out the class weights WC_c .

$$weight_c = \log \left(\frac{|Y|}{|y_c|} \right) \quad (2)$$

$$\begin{cases} WC_c = weight_c & weight_c > 1 \\ WC_c = 1 & otherwise \end{cases} \quad (3)$$

Based (2) and (3), $|Y|$ denotes the total number of samples in the data set and $|y_c|$ represents the count of samples attributed to class c . The provisions of this rule result in giving a weight of 1 to the majority class. Applying this rule assigns a weight of 1 to the majority classes. Simultaneously, it assigns a class weight approximately approaching 14 to the most infrequent instances. This method accords heightened significance to the least common cancer types while simultaneously recognizing the importance of the majority classes [24].

3. METHOD

3.1. Related work

The study [25] utilized various techniques such as random undersampling (RU), ROS, random oversampling combined with random undersampling (RURO), RU-SMOTE, and RU with adaptive synthetic sampling method (RU-ADASYN) across six cybersecurity datasets: these techniques were applied in conjunction with model classification using artificial neural network (ANN), DL, and ML; according to this research [24], comparing the techniques used are class-weight, oversampling, undersampling, and smote in binary classification; explores various synthetic oversampling methods and introduces a new oversampling algorithm that relies on the Mahalanobis distance [26]; demonstrated that in datasets with low dimensions, basic undersampling often achieves better performance compared to SMOTE [27]. A few of the latest studies

focusing on algorithm-level approaches include: analyzing the current deep-learning methods aimed at handling imbalanced class distributions [28] and creating a unique BalanceCascade-based kernelized extreme learning machine specifically tailored to tackle class imbalance issues [29]. Introduced a fresh approach to quantify imbalance by introducing a set of null-biased multi-perspective class balance metrics, expanding the notion of class balance accuracy to encompass additional performance metrics [30]; handled both the majority and minority classes separately and autonomously [31]. According to the mentioned research using multi-class classification in dealing with “Web” attack scenarios [32], [33]; utilized an additional convolutional neural network+long short-term memory (CNN+LSTM) model to conduct multiclassification experiments on the NSL-KDD dataset [34], [35].

3.2. Dataset

The dataset employed in the CICDoS 2019 competition originates from the Canadian Institute for Cybersecurity. DDoS attacks within the dataset encompass a combination of simulated and real-time instances, grounded in authentic planar capacitor (PCAP) data [36]. Additionally provided are the results of a CICFlowMeter-V3 network traffic analysis, which includes labeled streams arranged by timestamps, protocols, the data includes information on attacks, source and destination IP addresses, and ports, stored in a CSV file. The researchers used a single type of DDoS attack in their study. Due to LDAP’s reliance on the TCP protocol, this attack was executed at the application layer. There are 2,113,234 rows and 80 columns in this enormous text.

The dataset contains current and valid instances of DDoS attacks that occur often and are comparable to PCAP data seen in real-world scenarios. The unprocessed data, comprising the event logs (both Windows and Ubuntu) and network traffic PCAP for every machine, each of which has been saved as a CSV file. The collection is organized in a daily schedule. We systematically recorded raw data daily, encompassing the packet capture (PCAP) of the machine’s network traffic, as well as event logs sourced from both Ubuntu and Windows operating systems.

3.3. Data preparation

Hardware and software are required to analyze the dataset at this point. In the meanwhile, we used @Jupyter 6.5.4 software to analyze the dataset using virtual machine hardware that had the following specifications: the system configuration includes 16 CPUs, each featuring an Intel® Xeon® Gold 6134 processor with a clock speed of 3.20 GHz. It has a virtual processor count of 26, operates at a speed of 3.19 GHz, and is equipped with 200 GB of RAM. Figure 3 shows the architectural data processing flow;

- CICDDoS2019 input dataset: obtaining the necessary dataset is the initial stage in training, testing, and validating the suggested model.
- Preprocessing: the method of cleaning data involves eliminating characteristics with equal values and also eliminating noise from unimportant data [37], [38]. This data-cleaning procedure significantly impacts performance since handling the cleaned data will result in less noise and complexity. In selecting features, we drop features with an object data type that we consider does not have information important in model performance. For missing values and NaN, we use the mean technique to deal with them.
- Imbalance data: after analyzing the dataset, we found that the dataset was imbalanced and the file “LDAP.csv” included three labels: benign, LDAP, and Netbios. For multiclass, we label it [0,1,2]. We propose several techniques for imbalanced data solutions, namely ROS, RUS, and class-weight.
- Data split: following the acquisition of the new dataset, we conducted a partitioning, the dataset is partitioned, dedicating 80% to the training set and the remaining 20% to the testing set. The method of data splitting is employed to segment the dataset into subsets applicable at distinct stages of the data analysis process. Generally, data splits are executed before the training of ML models or the assessment of model performance.
- Model: the framework for extracting knowledge from the dataset we entered is a deep neural network. It typically consists of the following three main layers: the input layer is in charge of the data that it receives from the dataset. One node typically represents a single dimension or all of the features that are present in the dataset. The concealed layer, colloquially known as the hidden layer, serves as the intermediary stratum that accepts input data. Conversely, the output layer, positioned as the terminal stratum in the neural network architecture, receives the data transmitted from the preceding layer.
- Evaluation: the prescribed matrix comprises specific criteria that enable the evaluation of the DL model’s effectiveness in detecting DDoS attacks. These criteria encompass; i) correctness, representing the model’s overall accuracy; ii) precision, indicating the model’s ability to accurately identify instances of assaults; iii) recall, characterizing the model’s proficiency in identifying attacks relative to the total number of actual assaults; and iv) F-measure, also known as F1-score, serving as a harmonic mean that combines recall and accuracy. The mathematical formulations employed for the computation of these metrics are delineated in the subsequent equation [39].

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FN} \right)$$

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

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

$$F1 - Score = 2 \left(\frac{Precision \times Recall}{Precision + Recall} \right)$$

Where, TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

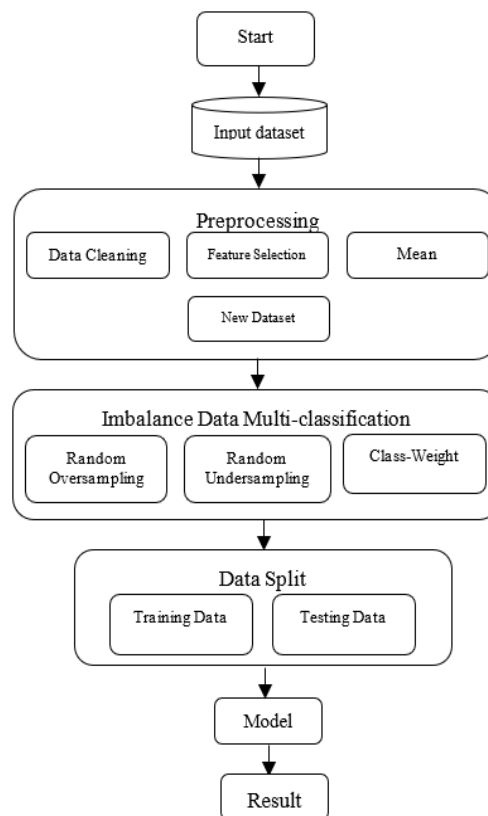


Figure 3. Data architecture processes

4. RESULTS AND DISCUSSION

In assessing performance concerning three imbalanced learning techniques specifically, ROS, RUS, and class-weight—applied to the multi-classification task. In the outcomes of this experiment, DDoS attacks on benign, LDAP, and Netbios tests are systematically employed for multiclassification evaluations to assess the proposed performance. Based on the findings and experimental results, most samples demonstrate accurate categorization into their respective classes. The observed diagonal positioning of samples suggests an enhancement in categorization performance.

On the contrary, the method's effectiveness undergoes a significant diminution when applied to a multi-classification testing scenario, as elucidated through a comparison of the illustrated results. This decrement is evident when subjecting the model to testing across these three specific techniques. This observation is further corroborated by the suboptimal performance exhibited by the suggested model in studies involving multi-classification contexts.

Confusion matrix is an evaluation tool used in ML to measure the performance of classification models. This matrix presents information about the predictions made by the model compared to the actual values in tabular form [39]. The confusion matrix table shows that our model evaluation identified most TP

cases, with TP values higher than TN, FP, and FN. Figure 4 shows the model’s ability to differentiate well between positive and negative cases in the dataset using the ROS technique. Figures 5 and 6 show the prediction errors or the number of examples misclassified to another class. In other words, the off-diagonal values show each class’s number of FPs and FNs. Figure 7(a) shows a graph comparing the level of accuracy vs epoch, Figure 7(b) shows loss vs epoch, Figure 7(c) shows loss vs epoch, and accuracy vs loss in ROS, Figure 8(a) also shows a comparison graph of RUS, accuracy vs epoch, Figure 8(b) shows loss vs epoch, Figure 8(c) shows accuracy vs loss, while Figure 9(a) shows a class-weight graph with a comparison of accuracy vs epoch, Figure 9(b) shows loss vs epoch, Figure 9(c) shows accuracy vs loss. For the three techniques, namely the random ROS, RUS, and class-weight techniques. This shows that the class-weight loss value is lower than that of the others and has a high accuracy value.

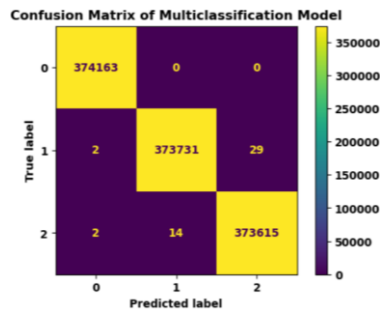


Figure 4. ROS

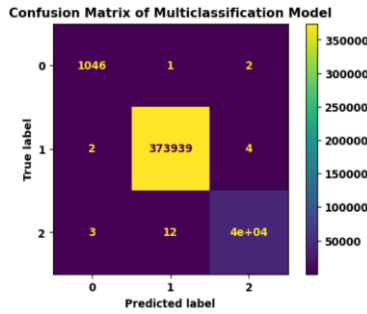


Figure 5. RUS

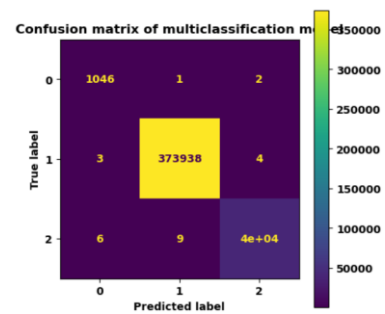
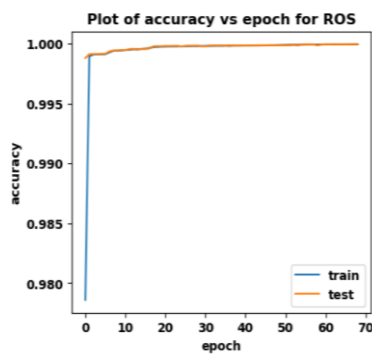
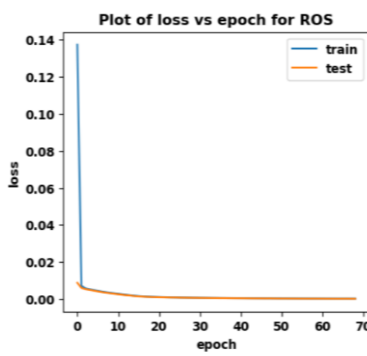


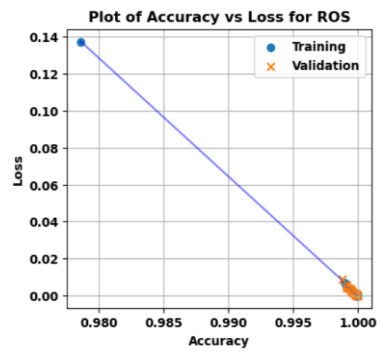
Figure 6. Class-weight



(a)

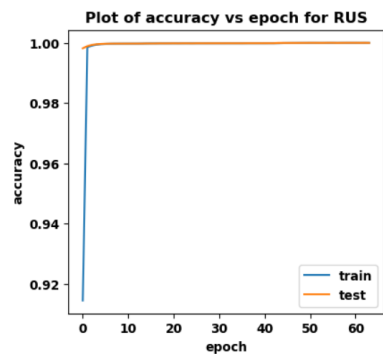


(b)

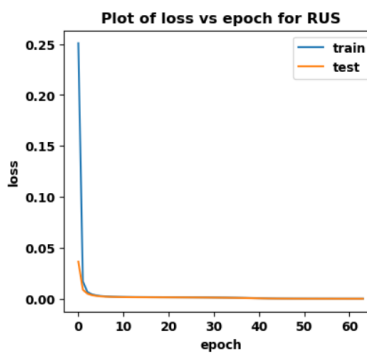


(c)

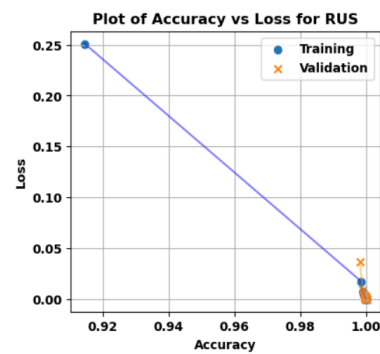
Figure 7. ROS; (a) accuracy vs epoch, (b) loss vs epoch, and (c) accuracy vs loss



(a)



(b)



(c)

Figure 8. RUS; (a) accuracy vs epoch, (b) loss vs epoch, and (c) accuracy vs loss

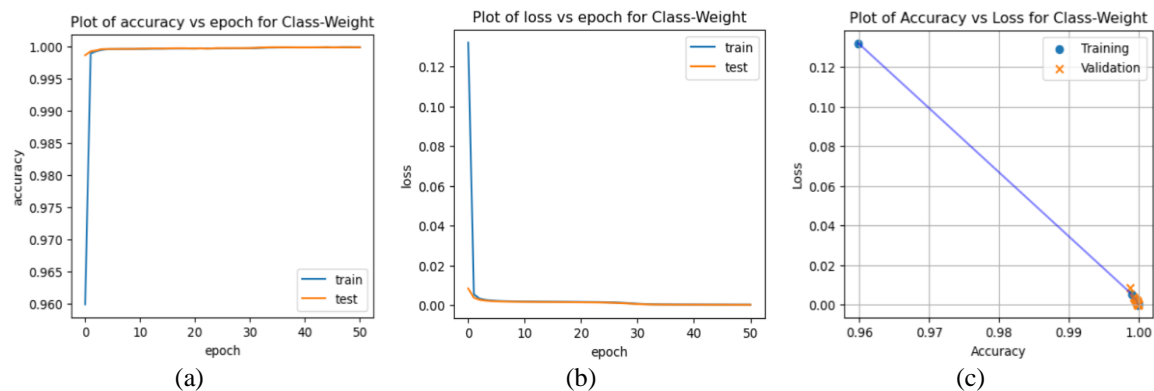


Figure 9. Class-weight; (a) accuracy vs epoch, (b) loss vs epoch, and (c) accuracy vs loss

5. CONCLUSION

From these results, conclusion can be drawn based on the work stages completed in the research. To overcome the problem of data imbalance for multiclassification in DDoS attacks with DL, the findings of comparing the performance of three imbalanced learning algorithms provide good accuracy performance in all three categories, namely benign, LDAP, and Netbios. Our results show that the largest loss value is RUS compared to ROS and class-weight, while the smallest loss value is class-weight in the multiclass setting regarding resolving unbalanced data and ROS has a better TP value. For future research, techniques such as AUC-ROC can be calculated using approaches such as one-vs-one (OvO) or one-vs-all (OvA) in evaluating multiclassification models, with a ML model approach and utilizing all existing features in the dataset without having to remove these features.





REFERENCES

- [1] H. Lin, S. Cao, J. Wu, Z. Cao, and F. Wang, "Identifying application-layer DDoS attacks based on request rhythm matrices," *IEEE Access*, vol. 7, pp. 164480–164491, 2019, doi: 10.1109/ACCESS.2019.2950820.
- [2] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.
- [3] Kaspersky, "TheKaspersky Q2 2020 DDoS attacks report," 2020. [Online]. Available: https://www.kaspersky.com/about/press-releases/2020_no-summer-vacation-ddos-attacks-tripled-year-on-year-in-q2-2020. (Accessed: Apr. 15, 2021).
- [4] M. Latah and L. Toker, "A novel intelligent approach for detecting DoS flooding attacks in software-defined networks," *International Journal of Advances in Intelligent Informatics*, vol. 4, no. 1, pp. 11–20, 2018, doi: 10.26555/ijain.v4i1.138.
- [5] H. Hanafi, A. Pranolo, Y. Mao, T. Hariguna, and L. Hernandez, "IDSX-Attention : intrusion detection system (IDS) based hybrid MADE-SDAE and LSTM-Attention mechanism," *International Journal of Advances in Intelligent Informatics*, vol. 9, no. 1, pp. 121–135, 2023, doi: 10.26555/ijain.v9i1.942.
- [6] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: a survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, 2021, doi: 10.1145/3448291.
- [7] P. Lou, Y. Yang, and J. Yan, "An anomaly detection method for cloud service platform," *ICMLT '19: Proceedings of the 2019 4th International Conference on Machine Learning Technologies*, Nov. 2020, vol. 8, no. 1, pp. 42–48, doi: 10.1145/3340997.3341005.
- [8] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016, doi: 10.1145/2907070.
- [9] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, "Imbalanced classification with multiple classes," *Learning from Imbalanced Data Sets*, pp. 197–226, 2018, doi: 10.1007/978-3-319-98074-4_8.
- [10] A. C. Tan, D. Gilbert, and Y. Deville, "Multi-class protein fold classification using a new ensemble machine learning approach," *Genome informatics*, vol. 14, pp. 206–217, 2003, doi: 10.11234/gi1990.14.206.
- [11] T. W. Liao, "Classification of weld flaws with imbalanced class data," *Expert Systems With Applications*, vol. 35, no. 3, pp. 1041–1052, 2008, doi: 10.1016/j.eswa.2007.08.044.
- [12] P. Santos, J. Maudes, and A. Bustillo, "Identifying maximum imbalance in datasets for fault diagnosis of gearboxes," *Journal of Intelligent Manufacturing*, vol. 29, no. 2, pp. 333–351, 2018, doi: 10.1007/s10845-015-1110-0.
- [13] N. Zarinabad, M. Wilson, S. K. Gill, K. A. Manias, N. P. Davies, and A. C. Peet, "Multiclass imbalance learning: Improving classification of pediatric brain tumors from magnetic resonance spectroscopy," *Magnetic Resonance in Medicine*, vol. 77, no. 6, pp. 2114–2124, 2017, doi: 10.1002/mrm.26318.
- [14] T. Sun, L. Jiao, J. Feng, F. Liu, and X. Zhang, "Imbalanced hyperspectral image classification based on maximum margin," in *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 522–526, Mar. 2015, doi: 10.1109/LGRS.2014.2349272.
- [15] P. Pramokchon and P. Piamsa-nga, "Reducing effects of class imbalance distribution in multi-class text categorization," *Recent Advances in Information and Communication Technology*, Springer, Cham, vol. 265, pp. 263–272, 2014, doi: 10.1007/978-3-319-06538-0_26.
- [16] M. B. Abidine and B. Fergani, "A new multi-class WSVM classification to imbalanced human activity dataset," *Journal of Computers*, vol. 9, no. 7, 2014, doi: 10.4304/jcp.9.7.1560-1565.
- [17] W. Prachuabsupakij and N. Soonthornphisaj, "Clustering and combined sampling approaches for multi-class imbalanced data




- classification,” *Advances in information technology and industry applications*, vol. 136, pp. 717–724, 2012, doi: 10.1007/978-3-642-26001-8_91.
- [18] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, “AMDO: an over-sampling technique for multi-class imbalanced problems,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018, doi: 10.1109/TKDE.2017.2761347.
- [19] G. Menardi and N. Torelli, *Training and assessing classification rules with imbalanced data*, vol. 28, no. 1. 2014, doi: 10.1007/s10618-012-0295-5.
- [20] M. Bach, A. Werner, and M. Palt, “The proposal of undersampling method for learning from imbalanced datasets,” *Procedia Computer Science*, vol. 159, pp. 125–134, 2019, doi: 10.1016/j.procs.2019.09.167.
- [21] M. C. Untoro and M. A. N. M. Yusuf, “Evaluate of random undersampling method and majority weighted minority oversampling technique in resolve imbalanced dataset,” *IT Journal Research and Development*, vol. 8, no. 1, pp. 1–13, 2023, doi: 10.25299/itjrd.2023.12412.
- [22] F. Pedregosa *et al.*, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [23] N. García-Pedrajas, J. Pérez-Rodríguez, M. García-Pedrajas, D. Ortiz-Boyer, and C. Fyfe, “Class imbalance methods for translation initiation site recognition in DNA sequences,” *Knowledge-based systems*, vol. 25, no. 1, pp. 22–34, 2012, doi: 10.1016/j.knsys.2011.05.002.
- [24] R. V. Asundi, R. Prakash, and K. Kumar, “Class weight technique for handling class imbalance,” *BMS Institute of Technology and Management*, Jul. 2022.
- [25] S. Bagui and K. Li, “Resampling imbalanced data for network intrusion detection datasets,” *Journal of Big Data*, vol. 8, no. 1, p. 40537, 2021, doi: 10.1186/s40537-020-00390-x.
- [26] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, “Experimental perspectives on learning from imbalanced data,” *ICML '07: Proceedings of the 24th international conference on Machine learning*, 2007, vol. 227, pp. 935–942, doi: 10.1145/1273496.1273614.
- [27] J. M. Johnson, T. M. Khoshgoftaar, B. S. Raghuvanshi, and S. Shukla, “Survey on deep learning with class imbalance,” *Journal of Big Data*, vol. 187, no. 1, p. 104814, 2019, doi: 10.1186/s40537-019-0192-5.
- [28] B. S. Raghuvanshi and S. Shukla, “SMOTE based class-specific extreme learning machine for imbalanced learning,” *Knowledge-Based Systems*, vol. 187, p. 104814, 2020, doi: 10.1016/j.knsys.2019.06.022.
- [29] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.
- [30] I. Triguero, M. Galar, D. Merino, J. Maillou, H. Bustince, and F. Herrera, “Evolutionary undersampling for extremely imbalanced big data classification under apache spark,” *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 640–647, 2016, doi: 10.1109/CEC.2016.7743853.
- [31] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, “Towards detecting and classifying network intrusion traffic using deep learning frameworks,” *Journal of Internet Services and Information Security (JISIS)*, vol. 9, no. 4, pp. 1–17, 2019, doi: 10.22667/JISIS.2019.11.30.001.
- [32] R. Atefinia and M. Ahmadi, “Network intrusion detection using multi-architectural modular deep neural network,” *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3571–3593, 2021, doi: 10.1007/s11227-020-03410-y.
- [33] X. Li, W. Chen, Q. Zhang, and L. Wu, “Building auto-encoder intrusion detection system based on random forest feature selection,” *Computers & Security*, vol. 95, 2020, doi: 10.1016/j.cose.2020.101851.
- [34] P. Sun *et al.*, “DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system,” *Security and Communication Networks*, vol. 2020, 2020, doi: 10.1155/2020/8890306.
- [35] R.-H. Hwang, M.-C. Peng, V.-L. Nguyen, and Y.-L. Chang, “An LSTM-based deep learning approach for classifying malicious traffic at the packet level,” *Applied Science*, vol. 9, no. 16, 2019, doi: 10.3390/app9163414.
- [36] Canadian Institute for Cybersecurity, “DATASET CICDDOS2019,” 2019. [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html>. (Accessed: Mar. 10, 2021).
- [37] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, “Enhancing data analysis with noise removal,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 304–319, 2006, doi: 10.1109/TKDE.2006.46.
- [38] M. Jupri and R. Sarno, “Data mining, fuzzy AHP and TOPSIS for optimizing taxpayer supervision,” *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 18, no. 1, pp. 75–87, Apr. 2019, doi: 10.11591/ijeecs.v18.i1.pp75-87.
- [39] M. S. Rana, C. Gudla, and A. H. Sung, “Evaluating machine learning models for android malware detection - a comparison study,” *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, pp. 17–21, Dec. 2018, doi: 10.1145/3301326.3301390.

BIOGRAPHIES OF AUTHORS






Rahmad Gunawan     graduated with a Bachelor’s degree at Gunadarma University with a major in Information Management, a Master’s degree with Gunadarma University majoring in electrical telecommunication. And now works as a lecturer at the Faculty of Computer Science, Universitas Muhammadiyah Riau. With research interests in the field of machine learning algorithms and AI. He can be contacted at email: goengoen78@umri.ac.id.





Hadhrami Ab Ghani    received his Bachelor degree in Electronics Engineering from Multimedia University Malaysia (MMU) in 2002. In 2004, he completed his Masters degree in Telecommunication Engineering at The University of Melbourne. He then pursued his Ph.D. at Imperial College London in Intelligent Network Systems and Completed his Ph.D. in 2011. He can be contacted at email: hadhrami.ag@umk.edu.my.



Nurulaqilla Khamis    received her Bachelor degree in Electrical and Electronics Engineering from Universiti Tenaga Nasional in 2012. In 2015, she completed her Masters degree in Artificial Intelligence at Universiti Teknologi Malaysia. She then pursued her Ph.D. at Universiti Teknologi Malaysia in Artificial Intelligence and completed her Ph.D. in 2020. Her current research work focuses on machine learning, deep learning, and swarm intelligence optimization. She can be contacted at email: nurulaqilla@utm.my.



Hasanatul Fu'adah Amran    received her Bachelor's degree in Mathematics Education from Ahmad Dahlan University in 2016. In 2019, she completed her Master degree in Mathematics Education at Universitas Ahmad Dahlan. Now works as a lecturer at the Faculty of Computer Science, Universitas Muhammadiyah Riau. With research interests in applied mathematics, algorithms, and AI. She can be contacted at email: hasanatul@umri.ac.id.