On the Effectiveness of Random Undersampling for Intrusion Detection of Imbalanced Network Traffic

Phi Ngoc Nguyen[1], Tuan-Cuong Vuong[2], Trang Mai Xuan[2]⋆, Vu-Duc Ngo[3], Hung Tran[4], Huan Vu[1], and Thien Van Luong[1]

[1] Business AI Lab, College of Technology, National Economics University, Vietnam
[2] A2I Lab, Phenikaa School of Computing, Phenikaa University, Hanoi, Vietnam
[3] Hanoi University of Science and Technology, Hanoi, Vietnam
[4] MobiFone HighTech Center, MobiFone Corporation, Hanoi, Vietnam
[5] DatCom Lab, College of Technology, National Economics University, Vietnam
nguyenngocphi.bvp.neu@gmail.com, {cuong.vuongtuan, trang.maixuan}@phenikaa-uni.edu.vn, duc.ngo@mobifone.vn, {hung.tran, huanv, thienlv}@neu.edu.vn

Abstract. In the field of network intrusion detection systems (NIDS), training data are often severely imbalanced between normal traffic and attack classes, which negatively affects the performance of machine learning (ML) models. This study focuses on analyzing the impact of random undersampling on NIDS using the UNSW-NB15 dataset. The proposed framework includes data preprocessing, the application of random undersampling to the training set, and the training of conventional ML models. System performance is evaluated using Accuracy, Precision, Recall, and F1-score metrics, with particular emphasis on the detection capability of attack classes. Experimental results demonstrate that random undersampling significantly improves the F1-score in binary classification; however, its effectiveness remains limited in multiclass classification scenarios. Our method also outperforms baselines in binary classification tasks.

Keywords: Intrusion detection · UNSW-NB15 · Undersampling · Machine learning.

1 Introduction

Network intrusion detection systems (NIDS) play a crucial role in monitoring network traffic and detecting malicious activities in a timely manner [19]. Various machine learning (ML) algorithms have demonstrated promising performance in intrusion detection tasks, making ML-based NIDS an active research area [13]. Despite the advantages of machine learning-based NIDS, their performance is strongly affected by the characteristics of training data. One of the most critical challenges is the severe class imbalance commonly observed in network intrusion

⋆ Corresponding author.

datasets [11]. This imbalance causes ML models to be biased toward the majority class, leading to high overall Accuracy but poor detection performance for minority attack classes [2]. As a result, the primary intrusion detection capability of NIDS can be significantly degraded [15].

To mitigate the impact of imbalanced data, various data-level and algorithm-level techniques were proposed [4, 8]. Among these approaches, undersampling is a widely used data-level technique that reduces the number of majority-class samples to achieve a more balanced class distribution. Undersampling is attractive due to its simplicity and reduced computational cost [6]. However, the effects of undersampling on the performance and robustness of ML-based NIDS, particularly across different training sizes on modern datasets, remain underexplored.

Motivated by the above challenges and research gaps, this paper investigates the effectiveness of undersampling techniques in ML-based NIDS. The main contributions of this study are summarized as follows:

- We propose a data imbalance handling approach based on undersampling to improve the performance of ML models in NIDS. With this design on the UNSW-NB15 dataset [17], the proposed method harnesses the strengths of multiple machine learning classifiers, resulting in significantly enhanced detection performance compared with traditional binary classification methods.
- We conduct a comparative analysis of multiple ML models trained with different dataset sizes after applying undersampling on the UNSW-NB15 dataset [17], providing insights into the relationship between training data scale and detection performance.

The rest of this paper is organized as follows. Section II reviews related works. Section III describes the UNSW-NB15 dataset [17] used in this study. Section IV presents our methodology, including the random undersampling strategy and the applied machine learning models. Section V reports and discusses the experimental results. Finally, Section VI concludes the paper and outlines directions for future work.

2 Related Work

2.1 ML-based NIDS

Machine learning has been extensively applied to network traffic analysis for detecting anomalous behaviors as well as previously unseen zero-day attacks without requiring prior knowledge of attack signatures [1]. As reported in [22], ML-based intrusion detection is commonly formulated as a supervised learning problem, where classifiers are trained on labeled datasets and later employed to predict the classes of unseen samples. An ML-based NIDS typically consists of three fundamental stages, namely data preprocessing, model training, and testing or evaluation [2].

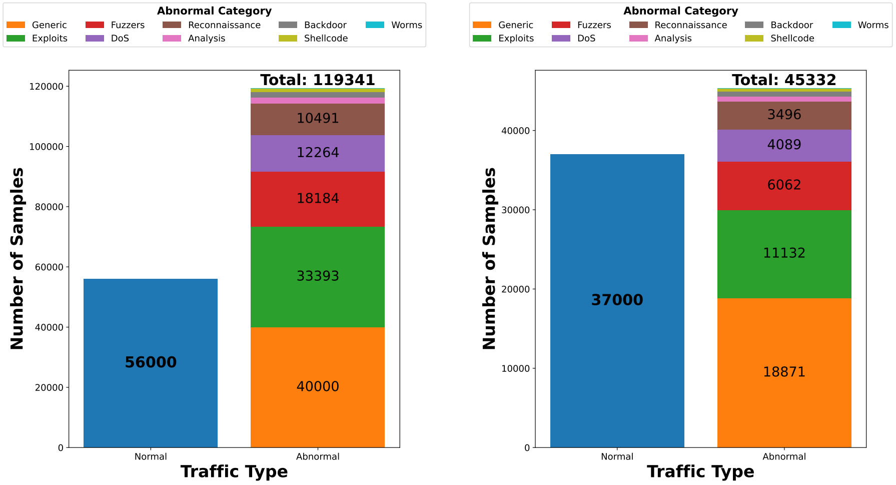
Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and ensemble models are widely applied in NIDS, and their effectiveness is strongly dependent on feature data processing steps such as normalization and dimensionality reduction. Experiments conducted on the UNSW-NB15 dataset [17] demonstrate that feature selection (FS) generally achieves higher detection Accuracy and lower computational cost when a sufficiently large number of features is retained, whereas feature extraction (FE) methods exhibit greater stability, lower sensitivity to changes in feature dimensionality, and superior performance in very low-dimensional settings, albeit at the expense of increased training cost [18]. The integration of principal component analysis (PCA) with ensemble learning has been shown to yield superior performance compared to FS approaches and single classifiers on the real-world UAV-ID dataset [7]; nevertheless, these gains come with higher system complexity and longer execution time [14]. In addition, optimizing the data processing pipeline for lightweight classifiers such as SVM and Gaussian Naive Bayes (NB), combined with data normalization and genetic algorithm–based parameter optimization, improves detection Accuracy, reduces detection latency, and significantly lowers false alarm rates [3].

2.2 Data imbalance in NIDS

Class imbalance is a critical issue in both cybersecurity applications and machine learning based intrusion detection systems [24]. A dataset is considered imbalanced when certain classes, particularly attack categories, are significantly underrepresented, making minority class detection more difficult and often degrading overall system performance [21].

One widely adopted strategy for handling class imbalance is oversampling minority classes. An online oversampling principal component analysis approach for anomaly detection was proposed in [12], targeting large-scale and online environments. By increasing the representation of minority-class instances, the method enabled effective anomaly detection while reducing computational complexity and memory usage. In [16], a hybrid intrusion detection framework combining synthetic minority oversampling technique (SMOTE), cluster centers, and nearest-neighbor techniques was presented and evaluated on the NSL-KDD dataset [5] using 10-fold cross-validation, achieving satisfactory Accuracy with a low false alarm rate. A hybrid method integrating SMOTE with edited nearest neighbors and a deep neural network was introduced in [23] and evaluated on the NSL-KDD dataset.

Undersampling techniques have also been shown to improve NIDS performance. The combination of CatBoost with random oversampling and undersampling was investigated in [10], achieving an Accuracy of 91.95% for multiclass classification on the CSE-CIC-IDS2018 dataset [20]. A hybrid resampling strategy using random oversampling for minority classes and random undersampling for majority classes was applied in [9] on the NSL-KDD dataset, leading to improved detection performance. Moreover, a two-stage resampling approach combining SMOTE-based oversampling and TomekLink-based undersampling was



(a) Training dataset (b) Testing dataset

Fig. 1. Distribution of classes in the UNSW-NB15 dataset.

presented in [16], where classification using a Long Short Term Memory model achieved an Accuracy of up to 99%.

3 Description of UNSW-NB15 dataset

3.1 Overview of UNSW-NB15 dataset

The UNSW-NB15 dataset [17], developed by the Australian Centre for Cyber Security, provides realistic benign and malicious network traffic and addresses the limitations of earlier network intrusion datasets. The complete dataset contains approximately 2.5 million network flow records, comprising 1 normal class and 9 distinct attack categories, namely Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms.

In this work, we utilize a cleaned 10% subset of the UNSW-NB15 dataset [17], which consists of 175,341 training samples and 82,332 testing samples. Notably, the selected 10% dataset contains no missing (null) values and no duplicated records. In addition, several irrelevant features namely scrip (source IP address), sport (source port number), dstip (destination IP address), and dsport (destination port number) were removed. Consequently, the feature set was reduced from 49 to 45 features, including 41 numerical features and 4 nominal features.

3.2 Imbalance characteristics of the dataset

The UNSW-NB15 dataset exhibits a highly imbalanced class distribution in both the training and testing sets, as illustrated in the provided figures. Fig. 1(a) shows the class distribution of the training dataset, where normal traffic constitutes the largest portion of the data. Among the abnormal categories, Generic and Exploits account for the highest number of samples, followed by Fuzzers, DoS, and Reconnaissance. In contrast, attack types such as Analysis, Backdoor, Shellcode, and Worms are represented by a substantially smaller number of instances. A similar class distribution pattern is observed in the testing dataset, as shown in Fig. 1(b).

In both data splits, the binary setting reveals a clear imbalance between normal and abnormal traffic. Under the multiclass setting, attack samples are unevenly distributed across categories, resulting in the presence of a few dominant classes and several sparsely represented ones. The primary difference between the two datasets lies in the total number of samples, with the testing set containing fewer instances while maintaining comparable relative class proportions.

4 Methodology

4.1 Random undersampling

In our methodology, random undersampling is applied as the first step and is restricted to the training set. For both binary and multiclass classification tasks, balanced training datasets are constructed with total sample sizes of 1,000, 2,000, 5,000, 10,000, 20,000, 40,000, and 80,000 samples, enabling a systematic analysis of the impact of dataset scale on model performance.

The undersampling procedure randomly selects an equal number of samples from each class in the training set. In binary classification, both classes provide sufficient samples to meet the target sizes. In the process of class balancing in multiclass classification, where some classes may be underrepresented, the algorithm retains the maximum available samples for those classes to minimize information loss while maintaining a relatively balanced class distribution.

4.2 Feature engineering

This stage is meticulously applied to both the training dataset and the testing dataset to ensure that the model is evaluated on a feature distribution that remains congruent with its learning environment.

Label encoding: In the preprocessing stage, categorical attributes are transformed into numerical representations using a label encoding scheme to enable effective processing by machine learning models. This approach maps each categorical value to a discrete integer while preserving the original feature dimensionality, making it suitable for datasets with a large number of distinct categories.

The training of the label encoder follows a strict procedure to ensure experimental integrity. Specifically, the encoder is fitted exclusively on the training

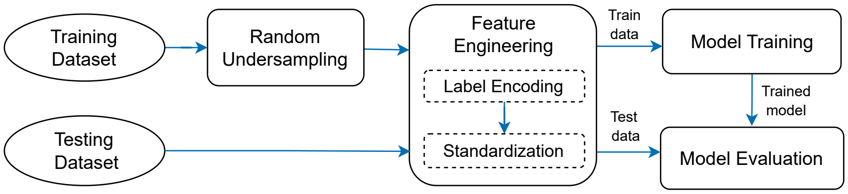


Fig. 2. Block diagram of the proposed NIDS.

set after class balancing, allowing the numerical mapping to be derived from a representative data distribution. This mapping is then fixed and consistently applied throughout the entire preprocessing pipeline. When applied to the test set, categorical values that were not observed during traiing are mapped to a predefined value of -1 to handle out-of-distribution cases.

Feature standardization: Following label encoding, numerical features are standardized to mitigate the effect of scale discrepancies among input variables. In this study, Z-score normalization is employed to rescale features to a common distribution, preventing features with larger magnitudes from dominating the learning process.

Formally, each feature value x is transformed as:

$$\mathrm{z} = \frac{\mathrm{x} - \mu}{\sigma}, \tag{1}$$

where μ and σ denote the mean and standard deviation of the corresponding feature, respectively. These parameters are estimated exclusively from the training set and subsequently applied to the test set. This separation prevents data leakage and preserves the rigor of the experimental setup.

4.3 Classification models

To evaluate the impact of the undersampling technique, we conduct experiments using several machine learning models commonly employed in NIDS studies. These models include DT, RF, k-Nearest Neighbors (KNN), Multi-layer Perceptron (MLP), NB, and Extreme Gradient Boosting (XGBoost). All models are trained with default hyperparameter settings, except for the MLP, where the maximum number of iterations (`max_iter`) is set to 300.

5 Experience results and discussion

5.1 Implementation setting

Computer configuration: The computer hardware configuration, operating system, and Python libraries used to implement the intrusion detection algorithms in this study are detailed in Table 1. The experimental environment is built upon widely adopted open-source libraries, which ensures stability during large scale data processing of the UNSW-NB15 dataset and enables accurate reproducibility of the experimental procedure.

Table 1. Computer configuration and software environment.

Unit	**Description**
Processor	13th Gen Intel i5-13500
RAM	64 GB
GPU	NVIDIA GeForce RTX 3060
Operating System	Ubuntu 24.04.2
Packages	Python 3.10, Scikit-learn, XGBoost

Evaluation metrics: The performance of the proposed method is evaluated on both binary and multiclass classification tasks using the test set of the UNSW-NB15 dataset, which consists of 82,332 samples. Four standard evaluation metrics are employed, namely Accuracy, Weighted Precision, Weighted Recall, and Weighted F1-score. Accuracy reflects the overall correctness of the classification results across all test samples. To address the class imbalance inherent in both binary and multiclass settings, Weighted Precision, Weighted Recall, and Weighted F1-score are adopted, where each class contribution is weighted by its number of samples. These metrics provide a more robust and fair assessment of the model's classification performance across different attack categories.

5.2 Binary classification

This subsection investigates the impact of training set size on the performance of different classifiers in the binary classification task. The comparative results across multiple training configurations are summarized in Table 2. For each column in the table, the bold values indicate the highest F1-score achieved by the corresponding classifier across different subset sizes. In particular, the red-highlighted value of 90.78% represents the overall best performance among all evaluated models and training settings, obtained by RF with 40,000 training samples. It can be observed that the performance does not increase monotonically with the training set size; instead, most classifiers achieve their peak performance on intermediate to large subsets rather than on the full dataset. In terms of stability, RF and XGBoost exhibit relatively small performance variations as the training size changes, whereas KNN and NB show more pronounced fluctuations, indicating lower robustness to variations in training data size.

To further examine the behavior of the best-performing model, Table 3 reports the detailed performance of RF under different training set sizes. The bold values in each row correspond to the best results achieved for the respective evaluation

Table 2. Comparison of F1-score between models for binary classification.

Samples	DT	RF	KNN	MLP	NB	XGBoost
1k	84.56	87.21	80.56	78.22	67.47	86.65
2k	86.76	86.69	83.74	85.72	70.82	88.66
5k	86.34	90.16	84.41	85.70	73.20	89.67
10k	85.64	90.21	85.05	88.03	72.43	89.43
20k	87.92	90.45	85.95	86.70	72.45	90.00
40k	**88.37**	**90.78**	85.94	88.74	73.07	90.23
80k	88.25	90.68	**86.29**	**88.77**	73.27	**90.45**
Full	86.29	86.81	83.97	86.91	**73.29**	86.95

Table 3. Performances of RF with different training sizes for binary classification.

Metric	1k	2k	5k	10k	20k	40k	80k	Full
Precision	88.49	90.25	90.64	90.76	90.97	**91.29**	91.14	88.87
Recall	87.43	89.79	90.25	90.30	90.53	**90.87**	90.75	87.13
Accuracy	87.43	89.79	91.56	90.30	90.53	**90.87**	90.75	87.13
F1-score	87.21	89.69	90.16	90.21	90.45	**90.78**	90.68	86.81

metrics. When trained with 40,000 samples, RF attains its highest F1-score of 90.78%, together with a Precision of 91.29%, a Recall of 90.87%, and an Accuracy of 90.87%, demonstrating a well-balanced detection capability. Compared with the full dataset setting, all evaluation metrics are consistently improved at this subset size. Furthermore, the performance remains relatively stable when the training set size increases from 20,000 to 80,000 samples, suggesting strong generalization ability. The inferior performance of the full dataset is mainly caused by severe class imbalance, while the balanced intermediate-sized subsets facilitate more effective decision boundary learning and yield superior performance.

Table 4. Comparison of the performances of the proposed method with other methods for binary classification.

Classifier	Precision	Recall	Accuracy	F1-score
FE+DT [18]	85.98	84.98	84.98	85.48
FE+RF [18]	85.85	80.03	80.03	82.55
FE+KNN [18]	86.39	84.99	84.99	85.69
FS+DT [18]	87.87	87.07	87.07	87.47
FS+RF [18]	85.74	80.95	80.95	83.27
FS+KNN [18]	87.08	85.98	85.98	86.53
Our method	**91.29**	**90.87**	**90.87**	**90.78**

To evaluate the effectiveness of the proposed method, several baseline models are constructed for comparison, including DT, RF, KNN classifiers combined with two different feature processing strategies in [18]. Specifically, FE-based on PCA is applied to DT, RF, and KNN (denoted as FE+DT, FE+RF, and FE+KNN), while FS techniques are integrated with the same classifiers (denoted as FS+DT, FS+RF, and FS+KNN). For both FE- and FS-based baselines, the reduced feature dimensionality is set to 8, as this configuration achieves the highest baseline performance.

Table 4 show that our proposed method consistently outperforms the reference method reported in [18] across all evaluation metrics. Specifically, it achieves the highest Precision, Recall, and Accuracy, all exceeding 90%, which indicates a strong capability in correctly detecting intrusion events while maintaining a low false alarm rate. Moreover, the proposed method attains a F1-score of 90.78%, representing a substantial improvement over the baseline approach.

5.3 Multiclass classification

The multiclass classification performance under varying training set sizes is evaluated using the F1-score, as summarized in Table 5. For all considered models, the highlighted values correspond to the highest F1-scores achieved within each column, while the red-highlighted value indicates the best overall performance across all models and training configurations. A clear trend can be observed where the F1-score consistently increases as more training data are incorporated. In particular, DT, RF, KNN, MLP, and XGBoost all achieve their maximum F1-scores when trained on the full dataset. Moreover, the gradual and monotonic performance improvement across increasing sample sizes suggests that the evaluated models exhibit stable behavior with respect to training data variation, without significant fluctuations or performance degradation.

Table 5. Comparison of F1-score between models for multiclass classification

Samples	DT	RF	KNN	MLP	NB	XGBoost
1k	68.31	70.76	59.94	66.24	46.72	71.17
2k	69.77	70.97	61.95	68.07	45.18	71.67
5k	71.19	71.61	66.90	69.81	41.68	72.76
10k	71.75	72.16	67.97	69.69	41.86	72.93
20k	72.04	72.44	68.76	70.16	41.26	73.24
40k	73.06	72.92	69.87	71.15	42.69	73.89
80k	73.87	73.56	70.58	71.97	43.26	74.50
Full	**76.13**	**77.79**	**73.74**	**76.46**	**46.72**	**78.24**

To further analyze the best-performing model, Table 6 presents detailed evaluation metrics of XGBoost across different training set sizes. Although the highest Precision is obtained at an intermediate subset of 10,000 samples, Recall, Accuracy, and F1-score steadily improve as the training set size increases and reach

Table 6. Performances of XGBoost with different training sizes for multiclass classification.

Metric	**1k**	**2k**	**5k**	**10k**	**20k**	**40k**	**80k**	**Full**
Precision	82.73	83.75	84.45	**88.15**	85.31	85.18	85.28	82.71
Recall	65.67	66.15	67.22	67.32	67.70	68.79	69.93	**76.50**
Accuracy	65.67	66.15	67.22	67.32	67.70	68.79	69.93	**76.50**
F1-score	71.17	71.67	72.76	72.93	73.24	73.79	74.50	**78.24**

their maximum values when the full dataset is used. This indicates that incorporating more data substantially enhances the model's ability to correctly identify samples across multiple classes. Compared with the results obtained using balanced subsets, the superior performance achieved with the full dataset highlights the data-intensive nature of multiclass classification, where a larger decision space and increased class overlap require more training samples to learn robust and discriminative decision boundaries.

6 Conclusion and future work

This study evaluates the effectiveness of random undersampling for addressing data imbalance in machine learning–based intrusion detection on the UNSW-NB15 dataset, considering both binary and multiclass classification across varying training data sizes. The results indicate that random undersampling substantially mitigates class imbalance effects in binary classification, leading to notable improvements in F1-score. In particular, the best binary classification performance is achieved when training on 40,000 samples using the RF model. Ensemble models, including RF and XGBoost, consistently demonstrate superior and stable performance across different training sizes. Experiments on the UNSW-NB15 dataset demonstrate that the proposed method outperforms baseline approaches including base models with existing feature extraction and feature selection techniques in binary classification tasks. In multiclass classification, although performance improves with increasing training data size, random undersampling alone is insufficient to yield comparable gains, reflecting the greater complexity of multiclass intrusion detection. Notably, the highest F1-score in the multiclass setting is obtained when training on the full dataset using the XGBoost model. However, the study is limited by its focus on a single sampling strategy, traditional machine learning models, and a single benchmark dataset. Future work will explore more advanced imbalance handling techniques and validate the proposed approach on additional datasets.

References

1. Abdelkhalek, A., Mashaly, M.: Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. Journal of Supercomputing **79**(10) (2023)

2. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F.: Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies **32**(1), e4150 (2021)
3. Alghushairy, O., Alsini, R., Alhassan, Z., Alshdadi, A.A., Banjar, A., Yafoz, A., Ma, X.: An efficient support vector machine algorithm based network outlier detection system. IEEE Access **12**, 24428–24441 (2024)
4. Bagui, S., Li, K.: Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data **8**(1), 6 (2021)
5. Dhanabal, L., Shantharajah, S.: A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering **4**(6), 446–452 (2015)
6. Hasanin, T., Khoshgoftaar, T.M., Leevy, J.L., Bauder, R.A.: Severely imbalanced big data challenges: investigating data sampling approaches. Journal of Big Data **6**(1), 1–25 (2019)
7. Hassler, S.C., Mughal, U.A., Ismail, M.: Cyber-physical intrusion detection system for unmanned aerial vehicles. IEEE Transactions on Intelligent Transportation Systems **25**(6), 6106–6117 (2023)
8. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)
9. Jiang, J., Wang, Q., Shi, Z., Lv, B., Qi, B.: Rst-rf: A hybrid model based on rough set theory and random forest for network intrusion detection. In: ICCSP 2018. pp. 77–81 (2018)
10. Jumabek, A., Yang, S., Noh, Y.: Catboost-based network intrusion detection on imbalanced cic-ids-2018 dataset. Journal of Korean Institute of Communications and Information Sciences **46**(12), 2191–2197 (2021)
11. Karatas, G., Demir, O., Sahingoz, O.K.: Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset. IEEE Access **8**, 32150–32162 (2020)
12. Lee, Y.J., Yeh, Y.R., Wang, Y.C.F.: Anomaly detection via online oversampling principal component analysis. IEEE transactions on knowledge and data engineering **25**(7), 1460–1470 (2012)
13. Li, Y., Wei, X., Li, Y., Dong, Z., Shahidehpour, M.: Detection of false data injection attacks in smart grid: A secure federated deep learning approach. IEEE Transactions on Smart Grid **13**(6), 4862–4872 (2022)
14. Luong, T.V., Pham, V.C., Le, T.T.H., Tran, X.N.: Robust intrusion detection for unmanned aerial vehicles: a pca-based feature extraction and ensemble learning approach. Cluster Computing **28**(5), 346 (2025)
15. Magán-Carrión, R., Urda, D., Diaz-Cano, I., Dorronsoro, B.: Improving the reliability of network intrusion detection systems through dataset integration. IEEE Transactions on Emerging Topics in Computing **10**(4), 1717–1732 (2022)
16. Mbow, M., Koide, H., Sakurai, K.: Handling class imbalance problem in intrusion detection system based on deep learning. International Journal of Networking and Computing **12**(2), 467–492 (2022)
17. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS). pp. 1–6 (2015)
18. Ngo, V.D., Vuong, T.C., Van Luong, T., Tran, H.: Machine learning-based intrusion detection: feature selection versus feature extraction. Cluster Computing **27**(3), 2365–2379 (2024)

19. Samy, A., Yu, H., Zhang, H.: Fog-based attack detection framework for internet of things using deep learning. IEEE Access **8**, 74571–74585 (2020)
20. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., et al.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**(2018), 108–116 (2018)
21. Wang, S., Yao, X.: Multiclass imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(4), 1119–1130 (2012)
22. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C.: Machine learning and deep learning methods for cybersecurity. IEEE Access **6**, 35365–35381 (2018)
23. Zhang, X., Ran, J., Mi, J.: An intrusion detection system based on convolutional neural network for imbalanced network traffic. In: ICCSNT 2019. pp. 456–460. IEEE (2019)
24. Zuech, R., Hancock, J., Khoshgoftaar, T.M.: Detecting web attacks using random undersampling and ensemble learners. Journal of Big Data **8**(1), 75 (2021)