JURNAL RESPONSIVE Teknik Informatika

Juni 2026 | Vol. 10 | No.01
E-ISSN : 2614-7602
DOI: 10.36352/jr.v10i01.1516

COMPARATIVE ANALYSIS OF CLASSIFICATION METHODS FOR CYBERATTACK DETECTION ON COMPUTER NETWORKS

Sherly Agustini[1*], Mustakim[2], Okta Veza[3]
[1,2]Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Ibnu Sina, Batam, Indonesia
[2]Universitas Muhammadiyah Papua
Email: *[1]sherly@uis.ac.id, [2]mustakim.anggaleha@gmail.com,

Abstract

The rapid growth of computer networks has increased the complexity and intensity of cyber threats, making machine learning based intrusion detection one of the most widely studied defense mechanisms. This study compares the performance of five classification methods Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes in detecting five categories of network activity: Normal, Denial of Service (DoS), Probing, Remote to Local (R2L), and User to Root (U2R). Due to limited access to sensitive real-world network traffic data, this study uses a small-scale simulated dataset of 500 samples generated programmatically using controlled statistical distributions to represent the characteristics of each category, including the class-imbalance condition commonly found in real network traffic. The data was split into 70% training and 30% testing using a stratified scheme and evaluated using accuracy, precision, recall, F1-score, and computation-time metrics. Results show that Decision Tree achieved the highest macro F1-score (85.96%) with 94.00% accuracy, slightly ahead of Naive Bayes (84.52% macro F1-score, 95.33% accuracy). Random Forest recorded the highest overall accuracy (96.67%), but its macro F1-score (84.06%) lagged due to low recall on the U2R class, which has very few samples. Feature-importance analysis indicates that srv_count, dst_host_count, and count are the main determinants for distinguishing attack categories. Naive Bayes and KNN recorded the fastest computation times, while Random Forest required the longest training time. The small dataset size causes performance estimates on minority classes (R2L and U2R) to be prone to fluctuation, so these findings should be regarded as a preliminary proof-of-concept study requiring further validation using a larger dataset or real-world network traffic data.

Keywords—intrusion detection; classification; machine learning; network security; simulated data

INTRODUCTION

The growing number of devices connected to the internet, including Internet of Things (IoT) devices, cloud infrastructure, and institutional information systems, has expanded the attack surface of computer networks. This condition is accompanied by an increase in the frequency and variety of cyberattacks, ranging from Denial of Service (DoS) that cripples services, probing or scanning activities that search for security vulnerabilities, to privilege-escalation attempts such as Remote to Local (R2L) and User to Root (U2R). Static, signature-based Intrusion Detection Systems (IDS) are increasingly unable to keep pace with new and evolving attack patterns, driving the development of adaptive machine-learning-based approaches as an anomaly-detection mechanism.

Most machine learning based IDS studies rely on benchmark datasets such as KDD Cup 99 or NSL KDD, which were recorded in the late 1990s to early 2000s (Agalit & Idrissi Khamlichi, 2024). On the other hand, access to real-world network traffic data within institutions is often restricted by privacy policies, infrastructure confidentiality, and the security risk should such data be leaked or misused. This condition motivates the need for an alternative approach to conduct preliminary exploration and testing of classification models using controlled, programmatically generated simulated data, before the model is further validated on real operational data.

This study aims to compare the performance of five classification methods commonly used in intrusion detection research Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes on a simulated network traffic dataset representing five categories of activity: Normal, DoS, Probe, R2L, and U2R. The simulated data is generated programmatically using different statistical distributions for each category, including an imbalanced class proportion as is typical of real network traffic, in which normal activity is far more dominant than attack activity, which occurs rarely, particularly U2R.

The main contributions of this study are: (1) providing a network traffic simulation framework that is replicable, transparent, and free from third-party copyright or privacy issues; (2) conducting a comparative evaluation of five classification algorithms using multi dimensional evaluation metrics, including accuracy, precision, recall, F1-score, and computation time; and (3) analyzing the contribution of each network traffic feature to the classification decision through a feature-importance approach.

LITERATURE REVIEW

Machine-learning-based intrusion detection has been widely studied across various network contexts, ranging from conventional networks to IoT and edge-computing environments. Stewart, Kolajo, and Daramola (2024), in a systematic literature review of machine-learning-based intrusion detection research, found that Support Vector Machine, K-Nearest Neighbor, Decision Tree, Random Forest, Naive Bayes, and Convolutional Neural Network are the algorithms most frequently applied for IDS, while also highlighting persistent challenges in handling class imbalance and adapting to evolving attack patterns.

Zhang, Jia, Wang, Wang, Liu, and Yang (2022) conducted a systematic comparison of seven algorithms, namely Decision Tree, Naive Bayes, SVM, Random Forest, XGBoost, Convolutional Neural Network, and Recurrent Neural Network, on the KDD Cup 99 and NSL-KDD datasets. The study concluded that ensemble-learning algorithms generally produced better detection results than single algorithms, while Naive Bayes, despite its relatively low accuracy on already-learned data, showed an advantage in training speed and generalization ability toward new, previously unseen attack types. This finding aligns with the accuracy-versus-efficiency trade-off also examined in the present study.

In a different domain, Bari, Yelamarthi, and Ghafoor (2023) applied SVM, Decision Tree, and KNN to detect intrusions on vehicle Controller Area Network (CAN) buses using several real-world vehicular datasets, reporting that the combination of these algorithms achieved very high detection accuracy with a minimal error rate. This study reaffirms that conventional classification algorithms, when combined with appropriate feature selection, remain competitive against computationally heavier deep-learning approaches.

Methodologically, Random Forest, an ensemble method based on decision trees, has been shown by Choubisa, Doshi, Khatri, and Hiran (2022) to provide a simple yet robust approach for intrusion detection in cyber security, owing to its resistance to overfitting and its capacity to handle high-dimensional network-traffic data. SVM, with its maximum-margin approach,

remains widely used for classification problems with non-linear decision boundaries via the kernel trick, as demonstrated by Kumari and Vurukonda (2024) in a grid-search-optimized SVM model for network intrusion detection in cloud environments. KNN, as an instance-based algorithm, has been shown to be effective for IoT intrusion detection when combined with feature selection (Mohy-Eddine, Guezzaz, Benkirane, & Azrour, 2023), while a single Decision Tree follows the decision-tree learning principle applied by Azam, Islam, and Huda (2023) in their comparative analysis of intrusion-detection models. Unlike these four methods, Naive Bayes assumes feature independence, giving it much lower computational complexity, but potentially reducing accuracy when this independence assumption does not hold for correlated network-traffic features, as also observed by Ige and Kiekintveld (2023) in their comparison of Bayesian variants for network intrusion detection.

Unlike prior studies that generally rely on public benchmark datasets, this study adopts a programmatically generated simulated-data approach. This approach is particularly relevant for institutions such as universities or small-to-medium organizations that do not yet have access to large volumes of real network-traffic data, but still require preliminary insight into potentially suitable classification algorithms before investing in operational data collection.

RESEARCH METHOD

3.1 Research Framework

This study was carried out through five main stages: (1) generation of simulated network traffic data, (2) data preprocessing, (3) splitting of the data into training and testing sets, (4) training and testing of five classification models, and (5) evaluation and comparative analysis of the classification results.

3.2 Simulated Dataset

The dataset used in this study is entirely synthetic and was generated using the NumPy library through controlled probability distributions (Gamma, Log-normal, Poisson, and truncated Normal), without copying, sampling, or deriving values from any public benchmark dataset (such as KDD Cup 99, NSL-KDD, CICIDS, or UNSW-NB15). Each network-activity category was generated with different distribution parameters to reflect general characteristics reported in the literature, for example DoS traffic marked by a very high connection count (count) within a short duration, or Probe traffic marked by a high diff_srv_rate ratio due to scanning many different services. The total dataset comprises 500 samples with 12 numeric features and 1 category label, as summarized in Table 1. The dataset size was intentionally kept relatively small to support a computationally lightweight preliminary exploration; the implications of this small sample size on the stability of the evaluation results are discussed further in the Discussion section.

Activity Category	Number of Samples	Percentage
Normal	272	54.40%
DoS (Denial of Service)	124	24.80%
Probe (Scanning)	61	12.20%
R2L (Remote to Local)	30	6.00%
U2R (User to Root)	13	2.60%
Total	500	100%

Table 1. Category Composition of the Simulated Network-Traffic Dataset

A total of 12 numeric features were generated to represent connection characteristics, including duration (connection duration), src_bytes and dst_bytes (bytes sent and received), wrong_fragment (number of malformed packet fragments), count and srv_count (number of connections to the same host and service within a given time window), serror_rate and rerror_rate

(connection error rates), same_srv_rate and diff_srv_rate (ratio of same/different services accessed), dst_host_count and dst_host_srv_count (number of connections at the destination-host level), and protocol_type_num (numerically encoded protocol type). To simulate the imperfect nature of real-world data, 2% of the labels in the dataset were randomly shuffled (label noise), so that the boundaries between classes are not perfectly separable.

3.3 Data Preprocessing

The data was split into a training set (350 samples, 70%) and a testing set (150 samples, 30%) using a stratified-split technique to preserve the proportion of each category in both subsets. For models sensitive to feature scale SVM, KNN, and Naive Bayes — feature standardization was performed using StandardScaler so that each feature has a mean of zero and a standard deviation of one. Decision Tree and Random Forest, being tree-based models, do not require feature standardization as they are not sensitive to data scale.

3.4 Classification Algorithms Compared

The five classification algorithms compared in this study, along with their parameter configurations, are as follows:

1. Decision Tree (DT): maximum tree depth limited to 12 levels to reduce the risk of overfitting.
2. Random Forest (RF): uses 200 decision trees (n_estimators=200) with a maximum depth of 14 levels.
3. Support Vector Machine (SVM): uses a Radial Basis Function (RBF) kernel with a regularization parameter of C=5.
4. K-Nearest Neighbors (KNN): uses k=5 nearest neighbors with the Euclidean distance metric.
5. Naive Bayes (NB): uses the Gaussian Naive Bayes variant, which assumes a normal distribution for each feature.

3.5 Evaluation Scheme

The performance of each model was evaluated using accuracy, precision, recall, and F1-score, both as macro averages (treating each class equally) and weighted averages (accounting for the proportion of samples in each class). The macro average was chosen as the primary comparison reference because it is more representative of a model's ability to detect minority classes such as R2L and U2R, which in practice are the critical attack categories to detect despite their rarity. In addition to classification accuracy metrics, this study also recorded computation time (training time and testing time) as a practical consideration for deployment in real-time intrusion-detection systems. All experiments were run using the Python programming language with the scikit-learn library on the same computing environment to ensure consistency in the computation-time comparison across models.

RESULTS AND DISCUSSION

4.1 Comparative Performance of the Models

Table 2 presents the evaluation results of the five classification models on the test data (150 samples) based on accuracy, macro precision, macro recall, macro F1-score, and training time. Results are ordered from the highest to the lowest macro F1-score.

Table 2. Performance Comparison of the Five Classification Models on the Test Data

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	Training Time (s)
Decision Tree	94.00%	87.11%	85.65%	85.96%	0.005

Naive Bayes	95.33%	85.27%	84.53%	84.52%	0.002
Random Forest	96.67%	94.98%	82.24%	84.06%	0.362
SVM (RBF)	94.67%	85.44%	77.25%	80.25%	0.006
KNN (k=5)	94.00%	96.83%	73.35%	78.41%	0.002

Contrary to the common assumption that ensemble methods such as Random Forest always outperform on every metric, the results on this small scale dataset show that Decision Tree actually achieved the highest macro F1-score (85.96%) with 94.00% accuracy, slightly ahead of Naive Bayes (84.52% macro F1-score, 95.33% accuracy). Random Forest recorded the highest overall accuracy (96.67%) and the highest macro precision (94.98%), but its macro recall was only 82.24%, lower than Decision Tree (85.65%) and Naive Bayes (84.53%). This discrepancy is mainly caused by Random Forest's failure to correctly identify most of the U2R samples in the test set, with a recall of only 25% (1 out of 4 U2R test samples correctly classified), whereas Decision Tree correctly identified 75% (3 out of 4 samples) of the same class. SVM and KNN recorded the lowest macro F1-scores among the five models, mainly due to low recall on the R2L and U2R classes.

This pattern is closely related to the relatively small size of the dataset (500 samples), which leaves very few test samples for the minority classes — only 9 samples for R2L and 4 samples for U2R. Under these conditions, a misclassification of just one or two samples can significantly shift the recall or F1-score of that class, making the model ranking based on macro metrics more susceptible to fluctuation than would be the case with a larger dataset. This also explains why Random Forest, which appears superior in terms of overall accuracy and macro precision, does not rank first when based on macro F1-score. In terms of computational efficiency, Naive Bayes and KNN recorded the shortest training times (0.002 seconds each), while Random Forest required the longest training time (0.362 seconds) — nearly 190 times longer than Naive Bayes — due to the process of building 200 decision trees in parallel.

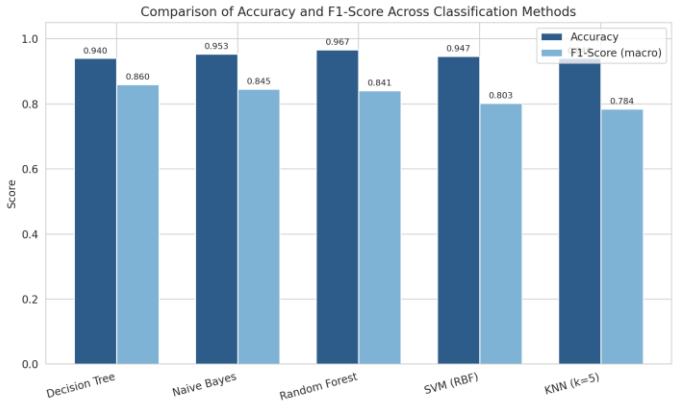

Figure 1. Comparison of Accuracy and F1-Score Across Classification Methods

4.2 Confusion Matrix of the Best-Performing Model

Figure 2 shows the confusion matrix of Decision Tree, the model with the highest macro F1-score in this experiment. This model classified the Normal, DoS, and Probe categories well, reflected in recall values of 97.56%, 94.59%, and 94.44% respectively, as summarized in Table 3. The R2L and U2R categories, which have the fewest test samples (9 and 4 samples respectively), produced lower recall values of 66.67% and 75.00%. Notably, the U2R recall for Decision Tree (75.00%) is far higher than for Random Forest (25.00%), but the U2R precision for Decision Tree is lower (60.00% versus 100.00%), indicating that this model is more aggressive in classifying samples as U2R, resulting in a number of misclassifications of other categories as U2R (false positives).

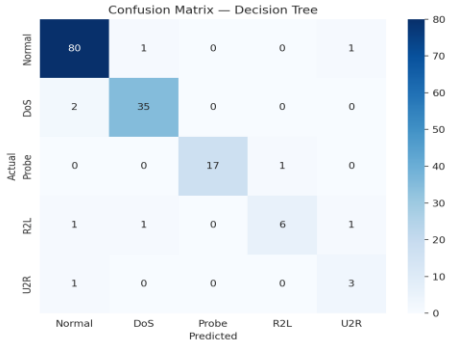

Figure 2. Confusion Matrix of the Decision Tree Model on the Test Data

Table 3. Per Class Classification Report for the Decision Tree Model

Class	Precision	Recall	F1-Score	Test Sample Count
Normal	95.24%	97.56%	96.39%	82
DoS	94.59%	94.59%	94.59%	37
Probe	100.00%	94.44%	97.14%	18
R2L	85.71%	66.67%	75.00%	9
U2R	60.00%	75.00%	66.67%	4

4.3 Feature Importance Analysis

Figure 3 presents the feature-importance scores based on the Random Forest model. Although Random Forest was not the model with the highest macro F1-score in this experiment, its feature importance is still used as an interpretability tool because it is built-in, relatively stable, and widely used in the intrusion-detection literature. The srv_count, dst_host_count, and count features, which represent the intensity of connections to a given service or host within a certain time window, were recorded as the most dominant determinants for distinguishing network-activity categories. This is consistent with the characteristics of DoS and Probe attacks, which are generally marked by a significant spike in connection count compared to normal traffic. The serror_rate feature also contributed substantially, consistent with the characteristics of DoS, which tends to produce a high connection-error rate due to excessive connection requests toward the target.

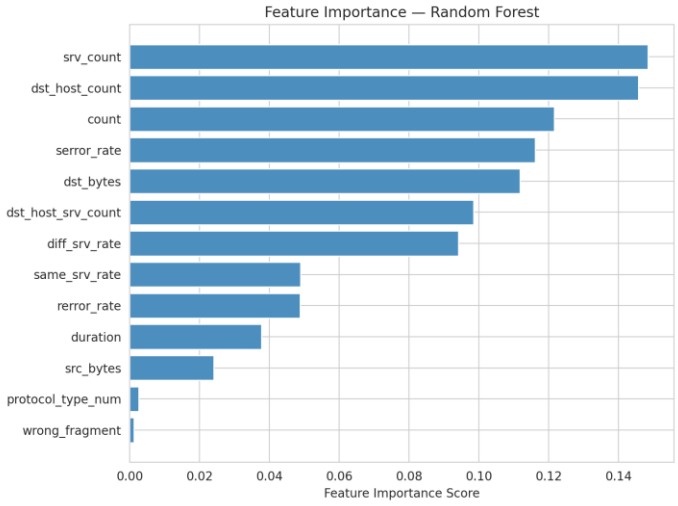

Figure 3. Feature Importance Based on the Random Forest Model

4.4 Computation Time Analysis

Figure 4 compares the training time of the five models. Naive Bayes and KNN showed the shortest training times because neither performs a complex iterative optimization process during training. Decision Tree and SVM were also relatively fast at this data scale. In contrast, Random Forest required a much longer training time, nearly 190 times longer than Naive Bayes, because it builds 200 decision trees in parallel even though the training set consists of only 350 samples.

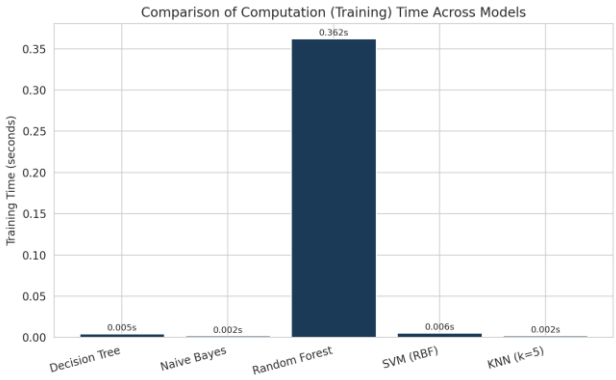

Figure 4. Comparison of Computation (Training) Time Across Models

4.5 Discussion

Overall, the results of this study show that the performance ranking of machine-learning models on a small-scale dataset can differ significantly depending on the evaluation metric used. If accuracy or macro precision alone is considered, Random Forest appears to be the best choice. However, when the macro F1-score, which is fairer toward minority classes, is taken into account, Decision Tree and Naive Bayes are actually superior because they better balance precision and recall on the R2L and U2R classes. This finding underscores the importance of not relying solely on overall accuracy when evaluating intrusion-detection systems, given that the rarest attack classes, such as U2R, are often the most critical categories to detect early.

The size of the simulated dataset in this study was intentionally limited to 500 samples to support a computationally lightweight preliminary exploration. As a consequence, the number of test samples for the minority classes is very limited only 9 samples for R2L and 4 samples for U2R — so that the recall and F1-score estimates for these classes have high variance and are prone to significant change from just one or two misclassified samples. Readers should interpret the performance ranking among models in this study as a preliminary indication rather than a definitive conclusion, given that the small test-sample size does not yet fully satisfy the statistical requirements for robust generalization. Further research using a larger dataset ideally at least several thousand samples with an adequate number of minority-class samples is needed to obtain more stable and reliable performance estimates.

The challenge of detecting minority classes such as U2R and R2L observed in this study is consistent with findings from various prior intrusion-detection studies (Zhang et al., 2022), which show that conventional classification algorithms tend to be biased toward majority classes when class imbalance is not explicitly addressed. For future research, applying data-balancing techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) (Elreedy, Atiya, & Kamalov, 2024) or cost-sensitive learning approaches has the potential to improve a model's ability to detect attack categories that are rare but critical in impact.

It should be emphasized that all data used in this study is simulated and was generated programmatically for methodological exploration and preliminary proof-of-concept testing of the five classification algorithms. The statistical characteristics of the simulated data were designed to represent general network-traffic patterns as described in the literature, but do not fully replace

the complexity and variation of real network-traffic data, which is influenced by network topology, service types, and actual user behavior within an institution. Therefore, the results of this study should be interpreted as a preliminary study that requires further validation using real network-traffic data before being applied to an operational intrusion-detection system.

CONCLUSION

This study compared the performance of five classification methods Decision Tree, Random Forest, SVM, KNN, and Naive Bayes in detecting five categories of network activity on a small-scale simulated network-traffic dataset of 500 samples. The experimental results show that Decision Tree achieved the highest macro F1-score (85.96%) with 94.00% accuracy, slightly ahead of Naive Bayes (84.52% macro F1-score, 95.33% accuracy), while Random Forest recorded the highest overall accuracy (96.67%) but lagged in macro F1-score (84.06%) due to low recall on the U2R class. Feature-importance analysis indicates that srv_count, dst_host_count, and count are the main determinants for distinguishing attack categories, while the computation-time analysis shows that Naive Bayes and KNN are the most efficient, whereas Random Forest requires the longest training time.

The R2L and U2R minority classes remain the main challenge for all models tested, with performance estimates prone to fluctuation due to the very limited number of test samples at this dataset size. Future research is recommended to validate these findings using a larger simulated dataset or real network-traffic data, to test data-balancing techniques such as SMOTE, and to explore deep-learning algorithms or hybrid approaches to improve detection capability for attack classes that are rare but have a critical impact on network security.

REFERENCES

1. Agalit, M. A., & Idrissi Khamlichi, Y. (2024). Optimization of intrusion detection with deep learning: A study based on the KDD Cup 99 database. International Journal of Safety and Security Engineering, 14(4), 1029–1038. https://doi.org/10.18280/ijsse.140402
2. Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. IEEE Access, 11, 80348–80391. https://doi.org/10.1109/ACCESS.2023.3296449
3. Bari, B. S., Yelamarthi, K., & Ghafoor, S. (2023). Intrusion detection in vehicle controller area network (CAN) bus using machine learning: A comparative performance study. Sensors, 23(7), 3610. https://doi.org/10.3390/s23073610
4. Choubisa, M., Doshi, R., Khatri, N., & Hiran, K. K. (2022). A simple and robust approach of random forest for intrusion detection system in cyber security. In 2022 International Conference on IoT and Blockchain Technology (ICIBT) (pp. 1–5). IEEE. https://doi.org/10.1109/ICIBT52874.2022.9807766
5. Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Machine Learning, 113, 4903–4923. https://doi.org/10.1007/s10994-022-06296-4
6. Ige, T., & Kiekintveld, C. (2023). Performance comparison and implementation of Bayesian variants for network intrusion detection. arXiv preprint arXiv:2308.11834. https://doi.org/10.48550/arXiv.2308.11834
7. Kumari, N. S., & Vurukonda, N. (2024). Support vector machine with grid search cross-validation for network intrusion detection in cloud. International Journal of Intelligent Systems and Applications in Engineering, 12(16s), 106–113.
8. Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network

intrusion detection model for IoT security using KNN classifier and feature selection. Multimedia Tools and Applications, 82(15), 23615–23633. https://doi.org/10.1007/s11042-022-13809-6

9. Stewart, D., Kolajo, T., & Daramola, O. (2024). Machine learning for intrusion detection systems: A systematic literature review. In K. Arai (Ed.), Proceedings of the Future Technologies Conference (FTC) 2024, Volume 1 (Lecture Notes in Networks and Systems, Vol. 1154). Springer. https://doi.org/10.1007/978-3-031-73110-5_42
10. Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative research on network intrusion detection methods based on machine learning. Computers & Security, 121, 102861. https://doi.org/10.1016/j.cose.2022.102861