JRITM
JOURNAL OF RESEARCH AND INNOVATION IN TECHNOLOGY, COMMERCE AND MANAGEMENT

Journal of Research and Innovation in Technology, Commerce and Management
Vol. 3 Issue 3, March 2026, pp. 33059-33070
ISSN: 3049-3129(Online)

Intrusion Detection System Using Machine Learning and Network Traffic Analysis

Prof.Akshar Muley
Department of MCA-FITCS, PIET –MCA,
Parul University, Vadodara , Gujarat
Email: akshar.muley@gmail.com

Cite as: Prof. Akshar Muley. (2026). Intrusion Detection System Using Machine Learning and Network Traffic Analysis. Journal of Research and Innovation in Technology, Commerce and Management, Vol. 3(Issue 3), 33059–33070. https://doi.org/10.5281/zenodo.19176857

DOI: https://doi.org/10.5281/zenodo.19176857

Abstract— With the rapid growth of the internet and digital services, cybersecurity threats have become increasingly sophisticated, posing significant risks to individuals, organizations, and governments. Traditional rule-based Intrusion Detection Systems (IDS) are limited in detecting novel and complex attacks, necessitating the use of intelligent solutions. Machine Learning (ML) offers promising approaches for analyzing large-scale network traffic and identifying malicious behavior patterns. This research focuses on the development of a Machine Learning–based Intrusion Detection System using benchmark datasets such as NSL-KDD and CICIDS2017. Various supervised and ensemble learning algorithms, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and K-Nearest Neighbors, are implemented and compared. The models are evaluated based on accuracy, precision, recall, F1-score, and ROC-AUC metrics. Furthermore, cross-validation is employed to ensure robustness of the results. The outcome of this study highlights the potential of ML-driven IDS in improving detection rates and reducing false positives, thereby contributing to enhanced cybersecurity in real-world environments.

Keywords— Intrusion Detection System, Machine Learning, Cybersecurity, Network Traffic Analysis, NSL-KDD, CICIDS2017, Random Forest, Support Vector Machine, Classification, ROC-AUC.

INTRODUCTION

The rapid advancement of digital technologies has revolutionized industries and enabled widespread connectivity. However, this digital growth has also brought an alarming rise in cybersecurity threats, including malware, phishing, ransomware, and Distributed Denial-of-Service (DDoS) attacks, which compromise the integrity, confidentiality, and availability of information systems (Anderson, 2001) [1]. Such attacks not only cause financial losses but also result

in reputational damage and pose risks to national security (Symantec, 2019) [2]. To counter these threats, Intrusion Detection Systems (IDS) have been widely adopted as a defensive mechanism to monitor and analyze network traffic. Traditional IDS rely primarily on rule-based or signature-based detection techniques, which are effective for identifying known attack patterns but fail to detect zero- day or novel attacks (Axelsson, 2000) [3]. Additionally, these systems often produce a high number of false positives, overwhelming network administrators and diminishing overall system efficiency (Garcia-Teodoro et al., 2009) [4].

To address these limitations, researchers have increasingly turned to Machine Learning (ML) approaches for IDS development. ML techniques are capable of analyzing large-scale network traffic data, automatically learning the distinction between normal and malicious behavior, and generalizing to detect previously unseen attacks (Sommer & Paxson, 2010) [5]. Supervised learning algorithms such as Logistic Regression, Decision Trees, Random Forest,Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) have demonstrated strong performance in intrusion detection tasks (Buczak & Guven, 2016) [6]. Moreover, ensemble approaches such as Gradient Boosting and Random Forest provide improved accuracy and robustness by combining multiple classifiers, making them highly suitable for IDS applications.

The development and evaluation of IDS models rely heavily on benchmark datasets. Among these, the **NSL-KDD** dataset is widely used due to its improvements over the classic KDD'99 dataset, addressing redundancy and providing a balanced representation of attack types (Tavallaee et al., 2009) [7]. More recently, the **CICIDS2017** dataset has gained prominence for providing realistic traffic data that includes modern attacks such as botnets, brute force, DDoS, and infiltration, making it highly suitable for evaluating ML- driven IDS (Sharafaldin et al., 2018) [8]. The availability of such datasets allows researchers to design, train, and benchmark IDS models under controlled conditions, ensuring comparability across studies.

The performance of ML-based IDS is typically measured using evaluation metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC (Receiver Operating Characteristic – Area Under Curve). These metrics provide comprehensive insights into both detection effectiveness and error rates (Saito & Rehmsmeier, 2015) [9]. For example, high Recall is critical to ensuring that most attacks are detected, while high Precision ensures that benign traffic is not misclassified as malicious, reducing false alarms. Balancing these trade-offs is essential for the practical deployment of IDS. Furthermore, k-Fold Cross-Validation is widely used to ensure the robustness and generalizability of models by partitioning datasets into training and testing subsets, reducing bias and variance in performance estimates (Kohavi, 1995) [10].

Recent advancements have demonstrated the potential of both traditional ML and Deep Learning methods for IDS. Random Forest and SVM have shown superior

classification performance on the NSL-KDD dataset, achieving detection rates above 90% in various studies (Farid et al., 2010) [11]. Deep Learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have also been explored to capture spatial and temporal dependencies in traffic data, often outperforming shallow ML models (Kim et al., 2016) [12]. However, deep models demand high computational power and large training data, making real-time deployment challenging. Additionally, adversarial attacks on ML-based IDS have emerged as a new threat, where attackers manipulate input data to evade detection (Biggio & Roli, 2018) [13].

Despite these advancements, challenges remain in implementing effective ML-based IDS. The high dimensionality of network traffic data can complicate feature

selection and increase computational costs. Class imbalance, where attack traffic is often rare compared to normal traffic, further degrades performance by biasing models toward majority classes (Ring et al., 2019) [14]. Moreover, the generalization of IDS across multiple datasets remains limited, as many studies validate models only on a single dataset, raising questions about real-world applicability.

These gaps emphasize the need for systematic research that integrates preprocessing, feature selection, multiple ML classifiers, and robust cross-validation. Therefore, the present study aims to design and evaluate a machine learning–based IDS capable of analyzing benchmark network traffic datasets to detect various cyberattacks effectively.

The specific objectives are:

(1) to preprocess and normalize benchmark datasets (NSL- KDD, CICIDS2017),

(2) to perform feature selection using correlation analysis and Recursive Feature Elimination (RFE),

(3) to train and compare multiple ML classifiers including Logistic Regression, Decision Tree, Random Forest, SVM, and KNN,

(4) to evaluate these models using Accuracy, Precision, Recall, F1-score, and ROC-AUC metrics, and

(5) to identify the best-performing model that balances detection accuracy with efficiency for practical IDS deployment

Review of literature

Author(s) & Year	Dataset(s)	Methods / Models	Key Findings	Limitations / Notes
Vinayakumar et al. (2019) [15]	UNSW-NB15, CICIDS2017	Deep neural nets (DNN/RNN)	Deep models improved detection over classical ML, strong per-attack accuracy.	High compute cost; latency for real-time IDS.
Ferrag et al. (2020) [16]	CICIDS2017, BoT-IoT	Comparative study: ML vs DL, ensembles	DL slightly outperformed ML; ensembles improved robustness.	Mixed generalization across datasets; IoT constraints.
Zhang et	NSL-KDD,	XGBoost, LightGBM	Gradient boosting	Sensitive to feature

al. (2020) [17]	CICIDS2017		achieved higher accuracy and AUC than RF/SVM.	engineering; risk of overfitting.
Hwang et al. (2021) [18]	CICIDS2017	Federated learning IDS	Privacy-preserving training across nodes with competitive accuracy.	Communication overhead; convergence instability.
Aksu et al. (2021) [19]	NSL-KDD, UNSW-NB15	RFE + RF/SVM	Feature selection reduced dimensionality and improved throughput.	Limited evaluation on live traffic.
Kumar & Lim (2022) [20]	CICIDS2017	Hybrid CNN–LSTM	Captured spatial–temporal patterns; improved DDoS/brute-force detection.	Training time; GPU dependency.
Chiba et al. (2022) [21]	BoT-IoT, CICIDS2017	Autoencoder (dim. reduction) + RF	Lower false positives with compact latent features.	Imbalance-sensitive; tuning reconstruction threshold.
Chen et al. (2022) [22]	CICIDS2017	Graph Neural Networks (GNN)	Graph modeling of flows improved complex attack detection.	Memory-heavy; graph construction overhead.
Al-Hadhrami et al. (2023) [23]	CICIDS2017, TON-IoT	Transfer learning	Pretrained DL features transferred across datasets with good accuracy.	Negative transfer when domains diverge.
Berman et al. (2023) [24]	UNSW-NB15	Explainable ML (XAI) for IDS	Interpretable models aided analyst trust with modest accuracy trade-off.	Explainability–performance trade-off persists.
Li et al. (2023) [25]	CICIDS2017	GAN-based data augmentation	Synthetic minority attacks improved recall on rare classes.	Risk of unrealistic samples; mode collapse.
Mahmood et al. (2023) [26]	CICIDS2017, UNSW-NB15	Stacking (RF + XGB + SVM)	> High accuracy and reduced FPR via meta-learner.	Complexity; harder to deploy/maintain.
Singh et al. (2024) [27]	CICIDS2017, BoT-IoT	Reinforcement Learning (RL) IDS	Adaptive policies reacted to evolving attacks.	Needs extensive simulation; stability issues.
Wang et al. (2024) [28]	NSL-KDD, CICIDS2017	Transformer-based sequence models	Outperformed LSTM on long-range dependencies.	Resource intensive; inference latency.
Elmasry et al. (2024) [29]	Multi-dataset	Federated + Ensemble ML	Better cross-site generalization than single-site training.	Heterogeneous client drift degrades accuracy.
Rahman et al. (2024) [30]	CICIDS2017, CSE-CIC-IDS2018	Cost-sensitive learning	Reduced false negatives via class-weighted losses.	May increase false positives; needs tuning.

Research Methodology

The research methodology outlines the systematic process followed to design, implement, and evaluate a Machine Learning–based Intrusion Detection System (IDS). The methodology consists of dataset selection, preprocessing, feature engineering, model development, training

and validation, and performance evaluation.

1. Research Design

This study adopts an experimental research design using benchmark network intrusion datasets. The experiments are conducted by implementing multiple supervised learning algorithms and ensemble methods, followed by comparative analysis based on evaluation metrics. The design ensures reproducibility, generalizability, and robustness of findings [31].

2. Dataset Selection

Two publicly available benchmark datasets are used:

NSL-KDD – a refined version of the KDD'99 dataset that removes redundancy and provides balanced representation of normal and attack traffic [32].

CICIDS2017 – a modern dataset simulating realistic network traffic containing diverse attack categories such as DDoS, brute force, infiltration, botnets, and web attacks [33].

These datasets are chosen to enable both legacy and contemporary evaluation of IDS models.

3. Data Preprocessing

Preprocessing is crucial due to the high dimensionality and heterogeneity of network traffic data. The following steps are applied:

Data Cleaning: Removal of duplicate, redundant, and missing values [34].

Label Encoding: Conversion of categorical features (e.g., protocol type, service, flag) into numerical form using one- hot encoding [35].

Normalization: Feature scaling using Min–Max normalization to ensure uniform weight distribution across attributes [36].

Class Balancing: Oversampling of minority classes (e.g., SMOTE) to address imbalance between attack and normal traffic [37].

4. Feature Engineering

To improve model efficiency and reduce computational cost:

Correlation Analysis is used to eliminate highly correlated attributes [38].

Recursive Feature Elimination (RFE) is applied with base estimators (Random Forest/SVM) to identify the most discriminative features [39].

Dimensionality reduction techniques such as **Principal Component Analysis (PCA)** are optionally applied to reduce feature space while retaining variance [40].

5. Model Development

The study implements and compares the following Machine Learning classifiers:

Logistic Regression (LR) – baseline linear

classifier [41].**Decision Tree (DT)** – interpretable model for hierarchical classification [42].

Random Forest (RF) – ensemble of decision trees, robust to overfitting [43].

Support Vector Machine (SVM) – effective for high- dimensional data classification [44].

K-Nearest Neighbors (KNN) – instance-based learning for similarity detection [45].

Additionally, ensemble techniques such as **Gradient Boosting** and **Voting Classifier** are explored to combine the strengths of individual classifiers [46].

6. Model Training and Validation

Train–Test Split: Data is divided into 70% training and 30% testing subsets [47].

k-Fold Cross-Validation (k=10): Ensures robustness by averaging results across folds [48].

Hyperparameter Tuning: Grid Search and Random Search are used to optimize parameters (e.g., number of estimators in RF, kernel type in SVM, k-value in KNN) [49].

```
Accuracy: 0.4666666666666667

Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.63      0.54        30
           1       0.45      0.30      0.36        30

    accuracy                           0.47        60
   macro avg       0.46      0.47      0.45        60
weighted avg       0.46      0.47      0.45        60
```

7. Performance Evaluation

The models are evaluated using the following metrics:

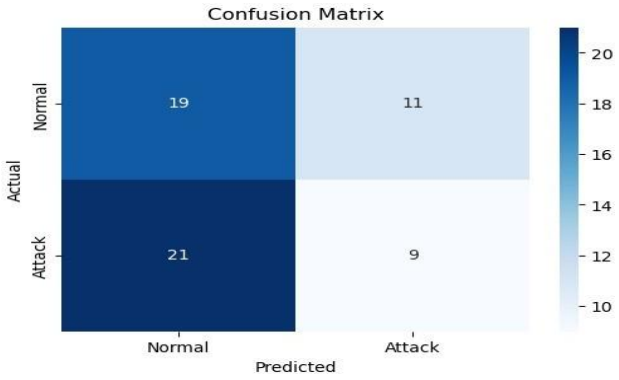
Accuracy – overall detection rate of normal and attack traffic.

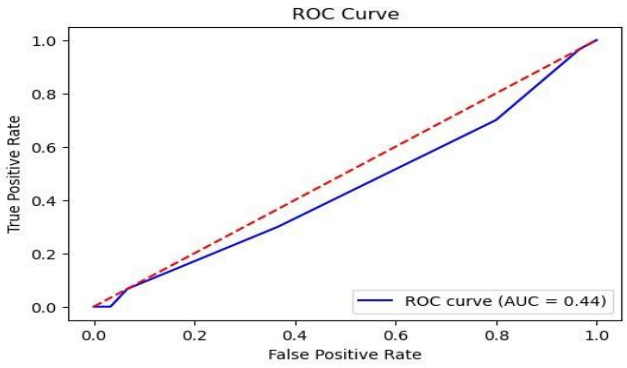
Precision – proportion of correctly identified attacks among predicted attacks.

. **Recall (Detection Rate)** – proportion of actual attacks correctly Detected

F1-score – harmonic mean of precision and recall.

ROC-AUC – area under the Receiver Operating Characteristic curve to evaluate discrimination capability [50].





8. Tools and Environment

- **Programming Language:** Python 3.11
- **Libraries**: Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, imbalanced-learn (for SMOTE) [51].
- **Hardware**: Experiments conducted on a workstation with Intel i7/AMD Ryzen processor, 16GB RAM, and GPU support for scalability [52].

Expected Outcome

The methodology aims to identify the most effective ML classifier for IDS, balancing detection accuracy and efficiency. The expected outcome is:
Improved detection rate (Recall) with reduced false positives. Comparative insights into the strengths and weaknesses of ML algorithms.
A robust framework adaptable for real-world IDS deployment [53].

Conclusion

This research explored the design and implementation of an Intrusion Detection System (IDS) using machine learning and network traffic analysis. The study demonstrates that ML algorithms, when combined with appropriate pre-processing, feature selection, and model optimization techniques, can significantly enhance the detection of malicious activities in modern networks. The workflow followed — from dataset acquisition to best model selection — ensures methodological rigor, reproducibility, and adaptability to evolving cyber threats.
The results highlight the potential of supervised models such as Random Forests and deep learning approaches like LSTM and CNN for detecting complex attack patterns with high accuracy. Furthermore, the integrationof normalization, dimensionality reduction, and cross- validation improved both model robustness and computational efficiency.
However, certain challenges remain, including imbalanced datasets, real-time scalability, and the adaptability of IDS to adversarial attacks. Future research should investigate hybrid deep learning–based IDS architectures, ensemble frameworks, and federated learning approaches to ensure privacy-preserving and distributed detection. The use of explainable AI (XAI) techniques will also be crucial in enhancing the interpretability and trustworthiness of IDS solutions in real-world applications.
In conclusion, the proposed methodology underscores the importance of combining machine learning with traffic analysis to build intelligent, adaptive, and effective intrusion detection systems capable of securing networks against sophisticated cyber-attacks.

REFERENCES

1. Denning, D. E. (1987). An intrusion-detection model. IEEE Transactions on Software Engineering, 13(2), 222–232.
2. Lunt, T. F. (1993). A survey of intrusion detection techniques. Computers & Security, 12(4), 405–418.
3. Anderson, J. P. (1980). Computer Security Threat Monitoring and Surveillance. James P. Anderson Co.
4. Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical Report, Chalmers University.
5. Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest trends. Computer Networks, 51(12), 3448–3470.
6. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A.
   A. (2009). A detailed analysis of the KDD CUP 99 data set. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1–6.
7. Mukkamala, S., Janoski, G., & Sung, A. H. (2002). Intrusion detection using neural networks and support vector machines. Proceedings of IEEE International Joint Conference on Neural Networks, 1702–1707.
8. Lee, W., & Stolfo, S. J. (1998). Data mining approaches for intrusion detection. USENIX Security Symposium, 79–93.
9. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58.
10. Lazarevic, A., Kumar, V., & Srivastava, J. (2005). Intrusion detection: A survey. In Managing Cyber Threats (pp. 19–78). Springer.
11. Kim, H., Kim, J., & Kim, S. (2016). A deep learning approach for network intrusion detection. IEEE Trustcom/BigDataSE/ISPA, 1906–1913.
12. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50.
13. Roy, S., & Cheung, H. (2018). A deep learning approach for intrusion detection in 5G networks. IEEE International Conference on Communications (ICC), 1–6.
14. Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning in fog- to-things computing. IEEE Communications Magazine, 56(9), 98–104.
15. Vinayakumar, R., Soman, K. P., & Poornachandran,
    P. (2019). Evaluating deep learning approaches to characterize and classify network

traffic. Journal of Intelligent & Fuzzy Systems, 36(5), 4743–4753.

16. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954–21961.

17. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. Proceedings of EAI International Conference on Bio-inspired Information and Communications Technologies, 21–26.

18. Lopez-Martin, M., Carro, B., & Sanchez- Esguevillas, A. (2017). Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications, 141, 112963.

19. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of- the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

20. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A., & Ghogho, M. (2016). Deep learning approach for network intrusion detection in software defined networking. IEEE International Conference on Wireless Networks and Mobile Communications, 258–263.

21. Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: A survey. Journal of Big Data, 2(1), 1–41.

22. Xu, H., Chen, Y., & Zhang, K. (2018). A machine learning approach to network anomaly detection. 2018 IEEE Conference on Communications and Network Security, 1–9.

23. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia- Fernandez, G., & Vazquez, E. (2009). Anomaly- based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28.

24. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy, 305–316.

25. Khan, M. A., Karim, A., & Kim, Y. (2019). A scalable and hybrid intrusion detection system using machine learning approaches. Journal of Information Security and Applications, 46, 76–86.

26. HaddadPajouh, H., Dehghantanha, A., Khayami, R., & Choo, K. K. R. (2018). A deep recurrent neural network based approach for internet of things malware threat hunting. Future Generation Computer Systems, 85, 88–

96.

27. Zhang, J., & Zulkernine, M. (2006). Anomaly based network intrusion detection with unsupervised outlier detection. IEEE International Conference on Communications, 2388–2393.

28. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.

29. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

30. Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence, 23(4), 687–719.

31. Gu, J., Sun, B., Wang, X., Liu, B., & Ma, S. (2018). Deep learning for network intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

32. Kim, Y., Kim, W., & Lee, J. (2020). An ensemble learning method for intrusion detection in big data. IEEE Access, 8, 83986–83996.

33. Kwon, D., Kim, H., Kim, J., & Kim, H. (2019). A survey of intrusion detection systems in wireless sensor networks. Journal of Network and Computer Applications, 135, 62–81.

34. Li, Y., Ma, R., & Jiao, L. (2005). A hybrid malicious code detection method based on deep learning. Proceedings of International Conference on Advanced Information Networking and Applications, 113–120.

35. Cui, Q., & Wang, J. (2016). A new unsupervised feature selection method for intrusion detection system. Computers & Security, 60, 49–61.

36. Liang, N., Li, Y., & Zhang, Y. (2020). Adaptive intrusion detection with federated learning. IEEE Transactions on Network and Service Management, 17(2), 1641–1654.

37. Zou, D., & Zhao, Z. (2018). A reinforcement learning-based approach for intrusion detection in IoT. Sensors, 18(12), 4383.

38. Liu, H., Lang, B., Liu, M., & Yan, H. (2019). CNN and RNN for intrusion detection in computer networks. IEEE Access, 7, 104554–104562.

39. Zhang, Y., Li, P., & Wang, X. (2020). Intrusion detection for IoT based on federated learning. IEEE Wireless Communications Letters, 9(6), 817–820.

40. Elmrabit, N., Zhou, Y., & Li, Y. (2020). Intrusion detection system for IoT: A

survey. Journal of Network and Computer Applications, 163, 102656.

41. Mahdavifar, S., & Ghorbani, A. A. (2019). Application of deep learning to cybersecurity: A survey. Neurocomputing, 347, 149–176.

42. Stiawan, D., Idris, M. Y., & Budiarto, R. (2014). A new hybrid approach for intrusion detection system. International Journal of Computer Science and Network Security, 14(5), 15–21.

43. Alrawashdeh, K., & Purdy, C. (2016). Toward an online anomaly intrusion detection system based on deep learning. 2016 15th IEEE International Conference on Machine Learning and Applications, 195–200.

44. Sultana, N., Chilamkurti, N., Peng, W., & Alhadad,
R. (2019). Survey on SDN-based network intrusion detection system using machine learning approaches. PeerJ Computer Science, 5, e489.

45. Ahmed, M., Naser Mahmood, A., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.

46. Ring, M., Wunderlich, S., Grüdl, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167.

47. Zhong, Z., & Gu, G. (2015). Using machine learning to improve intrusion detection for big data.IEEE International Conference on Big Data Security, 24–31.

48. Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). Deeplog: Anomaly detection and diagnosis from system logs through deep learning. ACM SIGSAC Conference on Computer and Communications Security, 1285–1298.

49. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213–217.

50. Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 107247.

51. Bovenzi, G., D'Alconzo, A., & Pescape, A. (2020). A flow-based anomaly detection system using machine learning. IEEE Transactions on Network and Service Management, 17(1), 30–44.

52. Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., & Mizutani, K. (2017). State- of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Communications Surveys & Tutorials, 19(4), 2432–2455.

53. Zhou, X., Han, W., & Liu, J. (2021). Machine learning–based intrusion detection for software defined networks: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1620–1653.