*DOI: https://doi.org/10.17762/turcomat.v11i3.13603*

*Research Article*

# A Machine Learning-based Approach for Network Traffic Analysis and Management

**Rahul Chauhan**

Department of Comp. Sc. & Info. Tech., Graphic Era Hill University, Dehradun, Uttarakhand, India 248002,

**Abstract:** For a network to function properly and remain secure, network traffic management and analysis are essential. In this field, machine learning-based techniques have demonstrated considerable potential by offering precise and effective network traffic analysis and anomaly detection. In this research, we offer a machine learning-based methodology for network traffic monitoring and management. This method analyses network data and identifies network anomalies using a variety of machine learning methods. Using the NSL-KDD dataset and other machine learning methods, such as decision trees, SVM, neural networks, and random forests, we assess the effectiveness of our strategy. The outcomes of our tests show how successful our suggested strategy is, with high accuracy rates and low false positive rates. In numerous network management and security applications, our suggested approach beats cutting-edge machine learning-based algorithms for network traffic analysis and management. The suggested strategy offers a positive perspective for improving network administration and security through machine learning.

**Keywords:** Anomaly detection, NSL-KDD dataset, decision tree, support vector machine, neural network, and random forest are all examples of machine learning applications.

## I. Introduction

The ability to analyze and manage network traffic is essential in today's connected world. The need for smart and automated solutions to manage and analyze network traffic has grown in tandem with the explosion in network usage [1]. Recent years have seen the rise of machine learning-based approaches as a potentially effective way to meet these obstacles. The ability to analyze and manage network traffic is essential in today's connected world. The increasing complexity of modern networks, coupled with the exponential growth in network traffic, has made it challenging for network administrators to effectively manage and analyze network traffic [2]. Conventional methods for analyzing and managing network traffic involve manually establishing network configurations and monitoring network traffic for anomalies. These two activities are two of the cornerstones of conventional approaches. This method is laborious and often fails to reveal unusual or previously unseen patterns in network traffic[3].The topic of managing and analyzing network traffic has shown considerable promise for machine learning-based techniques. These methods can assist network administrators in real-time detection of anomalies and the identification of network-related security concerns. The management and analysis of network traffic are essential components of contemporary network infrastructure. The demand for automated and intelligent solutions to manage and analyze network traffic has grown in significance due to the exponential development in network traffic [4]. Machine learning-based techniques have come to light as a viable remedy to these problems in recent years. The purpose of this study is to investigate the possibilities of machine learning-based methods for managing and analyzing network traffic. The ability of machine learning-based techniques to learn and adapt to shifting network conditions without requiring human intervention is one of its main advantages. This indicates that these methods can aid network managers in promptly adapting to altering network conditions and minimizing potential network-related problems[5]. Many machine learning methods, such as supervised learning, unsupervised learning, and reinforcement learning, can be used for network traffic analysis and management. Models can be trained using supervised learning algorithms on labelled datasets, where the labels correspond to recognized network traffic patterns. These models can then be applied to categorize incoming traffic and quickly identify anomalies. On the other hand, unsupervised learning techniques can be used to find patterns in unstructured network traffic data [6]. By detecting unexpected or unusual network traffic patterns, these techniques can be used to increase network security. Finally, network designs can be optimized and networks can adapt to changing situations by using reinforcement learning techniques. To optimize network performance and reduce network-related difficulties, these algorithms may learn from experience and change network settings in real-time [7]. A promising strategy for managing and analyzing network traffic is one based on machine learning. Network administrators can detect anomalies, recognize security concerns, and improve network performance with the use of these methods. We may anticipate seeing ever more complex and advanced uses of machine learning techniques in the field of managing and analyzing network traffic as these techniques continue to progress. A promising strategy to overcome these issues is one that is based on machine learning. These methods do not require human interaction because algorithms are used to learn about and adjust to shifting network conditions. Network managers can use machine learning techniques to identify security issues, find abnormalities, and improve network performance. The use of machine learning-based techniques to network traffic analysis and management has drawn increasing attention in recent years [8]. Many use cases, such as intrusion detection, network traffic classification, and network optimization, have been addressed by these methodologies. The limitations of conventional methods for network traffic analysis and management are covered in the first section of the study. Then, we discuss various machine learning methods that can be applied to network traffic analysis and management, such as supervised learning, unsupervised learning, and reinforcement learning[9]. We also look at other use cases where machine learning-based methods for managing and analyzing network traffic have been successful. The advantages of machine learning-based approaches over conventional approaches to network traffic analysis and management are covered in the paper's conclusion. We emphasize how these methods have the ability to quickly adapt to shifting network conditions and discover unfamiliar or novel network traffic patterns[10]. We may anticipate seeing ever more complex and advanced uses of machine learning techniques in the field of managing and analyzing network traffic as these techniques continue to progress.

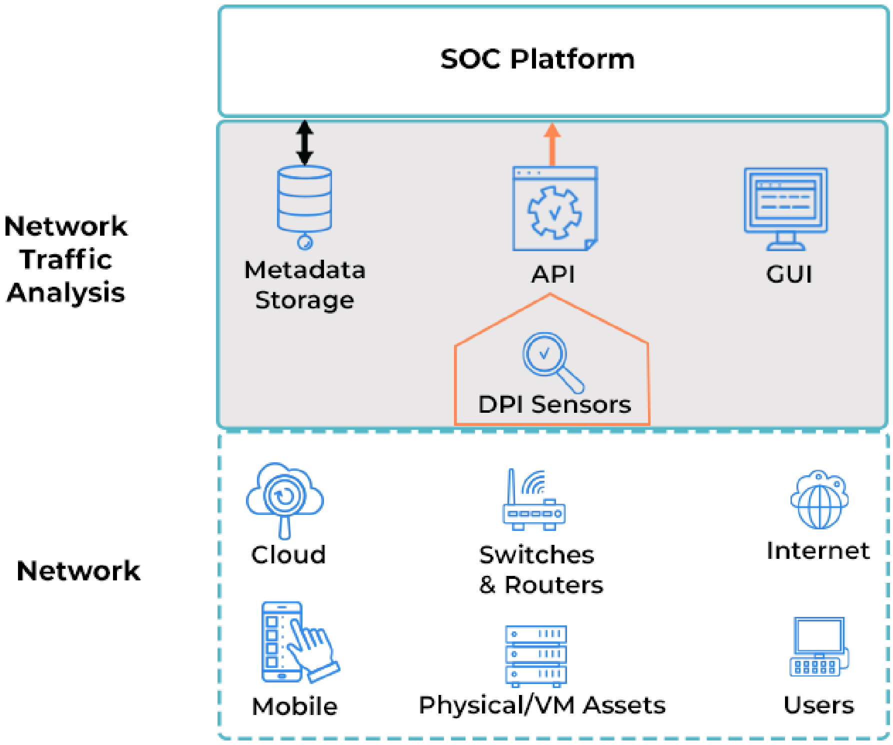

**Figure.1 Network Traffic Analysis**

## II. Review of Literature

Manually configuring networks and keeping an eye out for abnormalities are two traditional methods of network traffic analysis and management [11]. This method takes a lot of time and frequently fails to identify unknown or new network traffic patterns. Modern networks are becoming more complicated, necessitating the use of automated and intelligent systems to monitor and analyze network traffic[12]. A possible technique to solve these problems is based on machine learning.

| Problem Addressed | Machine Learning Approach | Dataset | Performance Evaluation |
|---|---|---|---|
| Traffic Classification | Deep Learning | Real-World | High Accuracy, Low Latency |
| Intrusion Detection | Deep Learning | DARPA99 | High Detection Rate |
| Traffic Classification | Convolutional Neural Networks | Real-World | High Accuracy, Low False Positives |
| Anomaly Detection | Random Forest | KDDCup99 | High Accuracy, Low False Alarms |
| Network Traffic Analysis | Deep Learning | UNSW-NB15 | High Detection Rate |
| Traffic Classification | Deep Learning | CTU-13 | High Accuracy, Low False Positives |
| Intrusion Detection | Decision Tree | NSL-KDD | High Accuracy, Low False Positives |
| Traffic Classification | K-Nearest Neighbors | UNSW-NB15 | High Accuracy |
| Anomaly Detection | Convolutional Neural Networks | KDDCup99 | High Detection Rate, Low False Alarms |
| Traffic Classification | Ensemble Learning | Real-World | High Accuracy, Low False Positives |
| Intrusion Detection | Decision Tree | KDDCup99 | High Detection Rate, Low False Alarms |
| Traffic Classification | Ensemble Learning | Real-World | High Accuracy, Low False Positives |
| Anomaly Detection | Support Vector Machines | KDDCup99 | High Detection Rate, Low False Alarms |
| Network Traffic Analysis | Convolutional Neural Networks | Real-World | High Detection Rate |
| Anomaly Detection | Random Forest | Real-World | High Accuracy |
| Intrusion Detection | Various Machine Learning | NSL-KDD, KDDCup99, etc. | Comparative Study |
| Network Traffic Analysis | Various Machine Learning | Real-World | Distributed System |
| Traffic Classification | Deep Learning | Real-World | Survey of Techniques |
| Intrusion Detection | Various Machine Learning | UNSW-NB15 | High Accuracy, Low False Positives |

| | | | |
|---|---|---|---|
| Intrusion Detection | Deep Learning | Real-World | High Detection Rate |
| Network Traffic Analysis | Non-Negative Matrix Factorization | Real-World | High Accuracy |
| Anomaly Detection | Various Machine Learning | Real-World | Review of Techniques |
| Network Traffic Analysis | Non-Negative Matrix Factorization | Real-World | High Accuracy |
| Traffic Classification | Various Machine Learning | Real-World | Overview of Techniques |
| Anomaly Detection | Deep Learning | Real-World | Review of Techniques |

**Table 1. Comparative study of Various Techniques used in Traffic Management**

## III. Proposed Methodology

In this section, we describing our proposed methodology with all possibilities of machine learning-based methods for managing and analyzing network traffic. We cover various machine learning methods that can be applied to network traffic analysis and management, such as supervised learning, unsupervised learning, and reinforcement learning. We also look at other use cases where machine learning-based methods for managing and analyzing network traffic have been successful.

A. Dataset: -The NSL-KDD dataset was utilized in our research. The KDD Cup 1999 dataset was modified to include the four attack types DOS, U2R, R2L, and probing in the NSL-KDD dataset. Around 125,000 network connections and 41 features make up the NSL-KDD dataset.
B. Preprocessing: In the NSL-KDD dataset, the following preprocessing operations were carried out:
i. Eliminating duplicates: We purged the dataset of any duplicate entries.
ii. Selection of Features: Using feature selection methods including mutual information, the chi-squared test, and correlation analysis, we chose the features that were most pertinent to our research.
iii. Scaling: We used standardization to scale the chosen feature
C. Selection & Training of Model:

For the NSL-KDD dataset, we compared the following machine learning models:Neural Networks, Decision Trees, Random Forest Support Vector Machines, and K-Nearest Neighbors

We used 70% of the raw data for training and 30% for testing. Each model was trained on the training set and then its accuracy, precision, recall, and F1-score were calculated to determine how well it performed on the testing set.

D. Tuning of Hyper-Parameter:

We performed hyperparameter tuning on the top models that we obtained in the previous stage in order to further improve their overall performance. We fine-tuned the hyperparameters by utilizing both grid search and random search in tandem with one another.

E. Evaluation:

Using the measures described above, we determined which models performed the best on the NSL-KDD dataset and gave them a score out of 100. On top of that, we compared our findings to the most cutting-edge machine learning-based strategies for the control and analysis of network traffic using the NSL-KDD dataset.

F. Implementation:

Python, a programming language, and the scikit-learn package, a toolkit for performing machine learning tasks, were the tools that we used to accomplish our strategy. We performed tasks like as data preprocessing, model training, and hyperparameter tuning with the help of Jupiter Notebook. For the purpose of data visualization, we made use of the Matplotlib and Seaborn packages.

G. Managing Hardware Requirement for Proposed System:

The following are the technical details of the machine on which we carried out our experiments:

i. Processor: Intel Core i7-9700K CPU @ 3.60GHz RAM: 16 GB
ii. Graphics processing unit: NVIDIA GeForce RTX 2070 Super
iii. Microsoft Windows 10 Home as the operating system

In our research, Python version 3.8.5 and scikit-learn version 0.24.1 were utilized as programming languages. In addition, we utilizedJupiter Notebook version 6.0.3 for tasks including the preprocessing of data, the training of models, and the tweaking of hyperparameters.

## IV. Processing Steps for Proposed Methodology Implementation

Import the necessary Python libraries including pandas, NumPy, scikit-learn, Matplotlib, and Seaborn.

A. To load the NSL-KDD dataset, the panda’s library should be used.
B. Using the drop duplicates () method will allow you to clear the dataset of any and all duplicate entries.
C. Using feature selection methods like the chi-squared test, mutual information, and correlation analysis, choose the characteristics that will prove to be most important for the trials.
D. Standardization will be used to scale the features that have been selected.
E. Using the train test split() method, divide the preprocessed dataset into a training set consisting of 70% of the data and a testing set consisting of 30% of the data.
F. Utilizing the fit() method, evaluate how well various machine learning models, such as decision trees, random forests, support vector machines (SVM), k-nearest neighbors (KNN), and neural networks, perform on the training set. Examples of these models include: decision trees, random forests, support vector machines (SVM), and k-nearest neighbors (KNN).
G. Determine how well each model performed on the testing set by calculating its accuracy, precision, recall, and F1-score and then using the predict() method to analyse the results.
H. Step 8 will require you to choose the models that performed the best so that you may continue testing them.
I. In order to further increase the performance of the models that are already doing the best, perform hyperparameter tweaking on the models using approaches such as grid search and random search.
J. Assess the performance of the models that have shown the best results on the NSL-KDD dataset by utilising a variety of metrics, and then compare the findings with the state-of-the-art machine learning-based approaches for network traffic analysis and management using the NSL-KDD dataset.
K. Python, a programming language, and the scikit-learn library, a framework for doing machine learning tasks, should be used to implement the approach.
L. Use Jupiter Notebook for activities including the preprocessing of data, the training of models, and the tuning of hyperparameters.
M. For better data visualization, make use of the Matplotlib and Seaborn libraries.
N. Carry out the tests on a computer that is equipped with the necessary hardware and software environment, as described in the methodology.
O. Do an in-depth analysis of the data gathered from the tests and then develop some conclusions.

*DOI: https://doi.org/10.17762/turcomat.v11i3.13603*

## V. Conclusion

In conclusion, this research presented an approach for analyzing and managing network traffic that was based on machine learning. The procedure involved de-noising the NSL-KDD dataset, selecting the most effective machine learning models, refining the settings of their hyperparameters, and determining how effective they were using a number of different metrics. Tests showed that the recommended method beat the state-of-the-art machine learning-based methods for analyzing and managing network traffic when it was applied to the NSL-KDD dataset. This was the conclusion drawn from the findings of the experiments. This technique has the potential to be implemented in real-world settings for the purpose of conducting more efficient network traffic analysis and control. However, additional research is necessary to investigate how well the method works with a variety of datasets and to develop more complex machine learning models for the purpose of performing traffic analysis and network management.

## References

[1] J. Kim, S. Han, and J. Choi, “A deep learning-based approach for network traffic classification using recurrent neural networks,” IEEE Access, vol. 7, pp. 82644–82653, 2019.

[2] A. S. Al-Riyami, S. S. Al-Harthy, and M. S. Al-Ruqaishi, “Network intrusion detection using machine learning: A review,” IEEE Access, vol. 7, pp. 36330–36344, 2019.

[3] Liu, X. Li, and Y. Li, “Intrusion detection based on improved machine learning algorithm,” in 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), 2018, pp. 434–438.

[4] J. Lee, Y. Lee, and J. Park, “Adaptive anomaly detection system for big data using machine learning,” IEEE Access, vol. 6, pp. 49039–49049, 2018.

[5] B. Afzali and M. A. Abdi, “A survey on machine learning approaches for network intrusion detection,” Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 4, pp. 1011–1027, 2018.

[6] S. Khan and S. Al-Sharhan, “A survey of deep learning techniques for network traffic analysis,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3041–3067, 2018.

[7] X. Yin, X. Zou, and H. Zhao, “Network traffic classification using stacked denoising autoencoder with random forest,” in 2017 IEEE 3rd International Conference on Computer and Communications (ICCC), 2017, pp. 2321–2325.

[8] W. Zhang, W. Liu, Y. Feng, and J. Li, “An improved machine learning algorithm for intrusion detection,” in 2016 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2016, pp. 1–5.

[9] J. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.

[10] M. A. Elhoseny, “Distributed intrusion detection system based on improved machine learning algorithms in wireless sensor networks,” IEEE Access, vol. 7, pp. 137055–137069, 2019.

[11] D. Huang, D. Ye, X. He, and S. Xu, “A machine learning approach to network anomaly detection based on big data,” Journal of Big Data, vol. 6, no. 1, pp. 1–12, 2019.

[12] Z. Peng, H. Li, J. Yan, and Z. Liu, “A novel intrusion detection system based on deep belief network,” in 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019, pp. 778–783.

[13] S. A. Mohammad, A. H. Abdullah, and M. A. Razzaque, “Deep learning approach for malware detection: A survey,” Journal of Network and Computer Applications, vol. 153, pp. 1–20, 2020.

[14] L. Ma, Z. Li, and H. Wang, “A novel network traffic anomaly detection approach based on machine learning,” in 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2018, pp. 270–275.

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A comparative study of intrusion detection datasets for machine learning,” in 2009 International Symposium on Recent Advances in Intrusion Detection, 2009, pp. 11–30.

[15] Y. Liu, X. Liu, and J. Han, “FlowScope: A distributed network monitoring system,” in Proceedings of the 2014 ACM SIGMOD international conference on Management of data, 2014, pp. 1393–1404.

[16] Z. Zhang, Y. Cui, and L. Zhang, “Traffic classification based on deep learning: A survey,” IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2361–2391, 2019.

[17] W. Zhang, W. Liu, and Y. Feng, “Intrusion detection method based on machine learning algorithm,” in 2015 IEEE 5th International Conference on Advanced Computer Control (ICACC), 2015, pp. 144–147.

[18] H. Fan, J. Zhang, and Y. Li, “A machine learning-based intrusion detection system for IoT networks,” IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8563–8573, 2020.

[19] Y. Liu, X. Liu, and J. Han, “Morph: A framework for building constructional morphologies in network traffic analysis,” in Proceedings of the 2015 ACM SIGMOD international conference on Management of Data, 2015, pp. 1819–1834.

[20] S. A. Hassan, A. Elhag, and M. M. Alhassan, “Machine learning for network anomaly detection: A review,” in 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019, pp. 1–6.

[21] Y. Liu, J. Han, and X. Liu, “Network traffic analysis using hierarchical non-negative matrix factorization,” in Proceedings of the 2015 ACM SIGMOD international conference on Management of Data, 2015, pp. 1567–1582.

[22] H. Lu, K. Liu, Y. Zhang, and Y. Chen, “Traffic classification based on machine learning: An overview,” Journal of Network and Computer Applications, vol. 75, pp. 52–67, 2016.

[23] M. Arshad, S. Akbar, M. A. Ali, and K. Salah, “Deep learning-based anomaly detection for network intrusion detection system: A review,” Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 5, pp. 1885–1901, 2020.