Network Anomaly Detection Using Machine Learning

Shreya[1], Harishma K V[2], Nisha K[3], Nishan M C[4], Varsha K[5]
[1,3,4,5]*Student, Srinivas Institute of Technology*
[2]*Assistant professor, Srinivas Institute of Technology*
doi.org/10.64643/IJIRTV12I12-206849-459

***Abstract*—The development of new Cyber Security threats will create challenges for Network Security which requires a dependable system that can track and study network irregularities to protect communication infrastructure. The use of traditional Anomaly Detection systems which base their detection methods on signature-based attacks will result in high False Positive rates and delayed threat detection because they cannot identify new and evolving attack patterns. The Automatic Anomaly Detection System that we developed in our invention needs to use Machine Learning algorithms for its network anomaly detection functions which will use manual packet input and continuous monitoring of network traffic to identify malicious activities. Our Network Anomaly Detection System uses multiple models including KNN RNN and LSTM which can classify Network Traffic into Normal or Anomalous states while we collect real-time network traffic through Scapy Python Network Scapy Libraries which allows us to monitor the network without external applications and tools. The Anomaly Detection Recommendation system provides end users with threat mitigation recommendations which include descriptions of new attacks that the system has detected. Our system testing results show that the LSTM Model outperformed all other models because it achieved better accuracy results with fewer false positives during anomaly detection compared to the models used in this system. The system aims to conduct.**

***Index Terms*—anomaly detection, network intrusion detection, machine learning, deep learning, LSTM, RNN, real-time network monitoring, network security**

I. INTRODUCTION

The modern computer networks face greater security threats because digital communication and cloud computing technologies continue to expand. The organization suffers major security breaches which affect data availability and confidentiality and integrity. Traditional intrusion detection systems (IDS) which depend on signature-based methods face limitations when they attempt to changing security threats. The system generates more false-positive results because of its design. Machine learning techniques provide an effective solution by identifying hidden patterns in network traffic data. The intelligent systems use their ability to learn from new data for better threat detection accuracy improvements. Machine learning-based intrusion detection systems achieve better network security results because they decrease false alarms while efficiently detecting both known and unknown cyberattacks. The current system needs intelligent security solutions which can automatically detect security threats through data analysis. Machine learning-based anomaly detection systems have attracted significant attention as prospective solutions to these challenges because they learn the typical network activity and are thus able to spot activity deviations which indicate malicious activity. Machine learning anomaly detection algorithms can identify new and zero-day assaults because they master network traffic behaviors and classify those behaviors as usual.

The detection performance of deep learning models improves when researchers use deep learning to model the temporal and sequential dependencies that exist within network traffic data. An artificial intelligence (AI)-based anomaly detection framework is developed to monitor live network traffic while it conducts packet analysis work. The Scapy library enables users to capture network packets in real time so they can perform basic packet analysis without needing additional software packages which include Tshark and Wireshark. The framework will use machine learning and deep neural network algorithms including K-nearest neighbor and recurrent neural network and long short-term memory to process the packet features which it will use to identify the attack that occurred through live network traffic. To further enhance the usability and interpretability of the framework by

average users and analysts, a web-based interface for reporting results was included. The framework includes a chatbot system which uses the Gemini API to provide users with contextual information about detected attacks and their corresponding prediction results. The proposed framework delivers a scalable solution together with an efficient method which protects modern network environments through enhanced cybersecurity measures. AI-driven analysis together with chatbot assistance enables users to understand better which helps them make decisions faster during security incidents. Advanced intrusion detection systems which protect network infrastructures against emerging cyber threats will be developed through the framework's implementation.

II. RELATED WORK

The development of various methods for detecting network anomalies and security breaches started with basic statistical methods and progressed to machine learning and finally reached modern deep learning techniques. The first Intrusion Detection Systems used both rule-based and signature-based detection techniques which restricted users to document attacks that had already been reported.

Researchers started to develop new methods which combine data-driven machine learning techniques with their existing knowledge because rule-based and signature-based methods had limitations and additional security weaknesses. Researchers began developing new techniques which combine data-driven machine learning methods with existing knowledge to detect and classify unknown security threats which include zero-day attacks. Researchers have investigated how traditional machine learning algorithms can be used to detect intrusions that occur on computer networks. The researchers focused their study on the KNN algorithm which belongs to the category of traditional machine learning algorithms because it uses feature vector similarities to classify network traffic and produces accurate results when applied to NSL-KDD structured data. The KNN algorithm requires excessive computational resources which results in decreased efficiency when used for large traffic detection and real-time network intrusion monitoring. The researchers created two new models which use Decision Trees and Random Forests to enhance both network traffic classification speed and decision process transparency. The Decision Tree and Random Forest Models can create multiple paths which result in better outcomes but they can also make the according to their original design because they can create multiple connections which lead to better results. Current is primarily focused on the use of live traffic collection tools to create systems that provide real-time Intrusion Detection capabilities. Scapy-based systems enable users to conduct real-time packet-level data analysis which helps them detect unusual network traffic patterns. Most real-time analysis systems lack both user interface to interact with users and system explanation functions, which creates obstacles for people with basic technical skills who need to understand how the system shows security threats.

Researchers are developing user-friendly AI systems which use KNNs and LSTMs and chatbots to deliver security information about attacks to users. The existing technology includes multiple improvements, but it lacks a standardized method for integrating Real-Time Traffic Monitoring with Sequential-Based Deep Learning models (RNN and LSTM) and User-Aided Explanation Mechanisms into one solution that provides complete functionality. The study presents a solution which combines KNNs with LSTMs and Scapy real-time traffic monitoring and bot-assisted detection of network anomalies, all presented through an explainable detection system.

The proposed research aims to address these limitations by developing an integrated anomaly detection framework which combines efficient traffic monitoring with intelligent attack detection and user-friendly interaction mechanisms. The framework uses real-time packet capturing through the Scapy library together with machine learning and deep learning algorithms to achieve better detection accuracy and faster response times. The integration of an AI-powered chatbot together with a web-based reporting interface improves threat detection interpretability, enabling both technical and non-technical users to understand network security incidents and implement suitable preventive measures.

III. PROBLEM STATEMENT

The Internet experiences multiple cyberattack types which include Denial of Service attacks Remote to Local attacks Probing attacks and User to Root attacks.

The current threat detection systems which depend entirely on signature-based methods fail to identify new threats and dynamic threats while they lack the capability to detect stealthy attack techniques. Network administrators face two major challenges because they need to monitor network attacks in real time but must deal with high false positive rates and their access to outdated databases which contain incorrect information.

The threat of cyberattacks requires organizations to implement lightweight solutions which can handle manual processing and network monitoring tasks in real-time without requiring extensive setup procedures. This proposal recommends creating an automated machine learning system which will detect abnormal network activities and determine the type of security threat an organization faces while providing guidance for improving network protection methods. The proposed solution requires three machine learning models K Nearest neighbor (KNN) Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) to produce actionable results through Scapy-based traffic recordings and an interactive recommendation system which delivers real-time interpretable results for efficient anomaly detection.

The proposed framework is designed to improve the accuracy, efficiency, and reliability of intrusion detection in modern network environments. The system uses traditional machine learning techniques together with advanced deep learning models to analyze network traffic patterns and detect both known threats and new attacks that occur in real time. The framework achieves better user experience through its combination of Scapy live packet capturing and interactive recommendation system which helps users understand security information. The approach helps organizations to build stronger cybersecurity systems while decreasing false alarms and enabling them to handle potential threats before they reach dangerous levels.

IV. PROPOSED SYSTEM

The system uses an intelligent machine learning based Anomaly Detection Framework which detects and classifies and analyzes all types of malicious activity from two data sources. The system development focused on machine learning methods which learned normal network activity patterns to improve threat detection through identification of abnormal behavior. The system retrieves network activity through two methods which start with the process of obtaining network activity data. Users create and input all packet parameters through Scapy to obtain packets which they enter manually during testing. Users have the option to capture all incoming network packets and analyze them with Scapy which enables them to examine packet header and payload information at the bit level. The system enables users to collect and analyze real-time network activity directly at their network boundary because it does not require extra packet analysis tools and it enables users to validate system performance during different operational modes of the system which include GUI and automated task execution.

The collected network data requires pre-processing work because network data needs to be transformed into a state of complete accuracy which meets established standards. The process of network data pre-processing will extract features from the complete network data set while it will convert all numerical features into standardized values and transform all categorical features which include protocol type and flags into encoded formats. The pre-processing activity aims to remove all background noise present in the original network data while it stops redundant feature attributes from being included in the machine and deep learning model training feature set.

The system implementation uses K-Nearest Neighbors and Recurrent Neural Networks and Long Short-Term Memory networks as its three different classification methods which operate on processed network traffic feature data. The KNN algorithm performs classification by measuring the similarity between an input instance and its nearest neighbors in the feature space, making it effective for identifying simple and well-defined anomalies.

RNN and LSTM models show the ability to understand sequential connections in traffic data which enables them to identify intricate attack patterns that evolve over time. The models study network activity patterns at different times which helps them find advanced unauthorized access attempts. The system identifies normal and harmful traffic while it classifies discovered anomalies into four primary attack categories which include Denial-of-Service (DoS) attacks, Probe attacks, Remote-to-Local (R2L) attacks, and User-to-Root (U2R) attacks. The

classification system uses learned feature patterns which are related to different attack types. In addition to the main prediction there is a prediction explanation which provides information about the prediction.

The system uses a recommendation module which provides context-based guidance to help users improve their security knowledge through their interactions and the system's ability to identify security breaches. The system presents detection results and alerts through an intuitive web-based interface which enables users without cybersecurity knowledge to understand the information. The system provides a feature that allows users to ask questions about unusual behavior detection which generates security recommendations to protect against possible threats. The system supports small-scale device deployment while enabling rapid response for extensive systems and handling automated traffic monitoring with machine learning and deep learning model classification and user interactive support features. The proposed system delivers an effective solution which provides real-time threat detection while improving network security in academic and business environments.

A. Algorithm Description and Workflow

The current section explains both the workflow and main functions of the proposed network anomaly detection system. The system description shows how network traffic gets processed and classified during both offline testing and real-time operation.

Network traffic can be captured using two approaches. The first method enables researchers to test different settings through manual input of testing parameters. The second method uses the Scapy library to enable continuous traffic monitoring through real-time data collection. The two input techniques work together within a single processing system which provides both reliable network data analysis and operational efficiency. During preprocessing, packet fields are validated, categorical attributes are transformed into numerical values, and numerical features undergo normalization to enhance model performance. The data preparation process eliminates unnecessary data elements while it transforms the collected information into the appropriate format needed for machine learning and deep learning training purposes.

The data goes to classification for analysis after the preprocessing work is finished. The current stage uses multiple models which include K-Nearest Neighbors (KNN) Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. The KNN model uses distance-based classification to detect basic anomalous patterns which makes it an effective tool for this task. The RNN and LSTM models use their ability to analyze network traffic data over time to detect both complex and time-changing anomalies. The system configuration process establishes which model to use based on the specific analysis requirements and deployment needs. The system generates three outputs after completing the classification process which includes a detection label and a confidence score and an attack category.

The Recommendation Part provides standard Mitigation Strategies whereas the Visualization Part shows current results from the system. Users can inter-act with the anomaly detection system via Chatbot (Natural Language Processing) and receive explanations of detected anomalies. This complete system enables precise anomaly detection which operates with quick response times during live operations and links all three stages of data collection and data analysis and user interaction through its various system components.

V. METHODOLOGY

The proposed methodology consists of four main stages: data collection, preprocessing, model training, and anomaly detection. The Scapy library enables two methods of obtaining network data which include manual packet input and real-time data capture. The collected data is then prepared through preprocessing steps such as validation, encoding of categorical attributes, and normalization of numerical features to ensure consistency.

The benchmark datasets contain normal and malicious traffic patterns which are used to train machine learning and deep learning models that include K-Nearest Neighbors and Recurrent Neural Networks and Long Short-Term Memory networks. The trained models receive packet features which are extracted from incoming packets during the detection phase. The system uses this analysis to determine whether traffic belongs to the normal category or the anomalous category which helps to identify possible network security threats. The system provides users with an easy-to-use method which detects anomalies with high accuracy and efficiency to monitor network traffic in

real time. Anomaly detection research progresses through online resources which include an automatic-response system that responds to detected anomalies and a chatbot which helps users with their network and related questions.

A. Dataset Description and Feature Analysis

The machine learning-based anomaly detection system requires high-quality data inputs for its operational effectiveness. The researchers developed a new system for anomaly detection which uses two data sets NSL-KDD and UNSW-NB15 as its foundation. The data sets include labeled samples of actual data which demonstrate standard behavior and multiple cyber-attack scenarios. The data set contains samples that show typical activities and multiple types of cyber-attacks which occur during regular operations. The dataset provides information about various network characteristics which include Protocol Type, Service Type, Flag Status, Packet Size, Connection Duration and various connection attributes. The first three features of the system which include protocol type and service type and flag status function as protocol-level attributes. The packet size and connection duration and connection-based statistics function as network usage behavioral features according to their nature. The datasets contain different attack types which include Denial-of-Service (DoS) and Probe and Remote-to-Local (R2L) and User-to-Root (U2R) attacks. The various attack types permit a complete assessment of how well the model performs. The team conducted feature analysis before training to find necessary attributes while they removed all redundant and unimportant features. The system standardized numerical features to achieve uniform measurements while it used suitable methods to transform categorical attributes into usable data.

The preprocessing activities establish better model performance through faster convergence rates and more accurate classification results which benefit distance-based algorithms and gradient-based models. The intrusion detection datasets experience a typical issue called class imbalance which occurs when regular traffic exceeds the actual detection of suspicious behavior. The research used balanced sampling methods during training to solve the problem because it helps to stop the model from learning only from the dominant class. The method enables equitable learning throughout all categories which enables the system to handle actual network traffic data that was obtained through the Scapy framework.

B. Data Collection

The training data is obtained from standard benchmark datasets such as NSL-KDD and UNSW-NB15. The system uses labeled records to distinguish between normal network behavior and different attack types.

C. Preprocessing

The system processes two types of data which include manually entered packet data and live traffic data that Scapy captures. The system cleans both data types to create machine learning-ready data which experts can use for their machine learning work. The data preparation process verifies packet field data while creating unified fields through the standardization of all numeric inputs and converting protocol and flag data into one common format. The model achieves improved performance because it can learn from clearer data which results in better accuracy and stable detection of anomalies in KNN, LSTM, and RNN model generated through this process.

D. Model Training

Algorithms used:

- K-Nearest Neighbors (KNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM)

Models are trained using supervised learning to classify attack types such as DoS, Probe, R2L, and U2R.

E. Prediction

The trained models which include K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks begin their prediction process after they receive preprocessed packet features as input. The models use their analysis of incoming traffic to identify whether the traffic patterns match normal operations or indicate a possible security threat.

The system needs to identify abnormal patterns which will lead it to establish four different attack types which include Denial-of-Service (DoS) and Probe and Remote-to-Local (R2L) and User-to-Root (U2R) attack types. The system provides a confidence score together with each prediction which shows how certain the system is about the detected attack.

VI. IMPLEMENTATION DETAILS

The Network Anomaly Detection System uses a modular software architecture design which enables both system expansion and its ability to function in real-time. Python was chosen as the main programming language because it provides a comprehensive collection of libraries which support both machine learning and network analysis work. The Scapy library functions as the tool for capturing network packets in real-time because it offers complete access to low-level packet data which enables effective traffic monitoring.

The system uses prepared datasets which researchers stored for future model testing in their experiments. The K-Nearest Neighbors (KNN) algorithm needs its nearest neighbor count to be determined through testing procedures. Deep learning frameworks enable developers to create Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models. All models undergo training through supervised learning methods which use labeled network traffic data for their training process. The model hyper-parameters which include learning rate and hidden layer(s) and sequence length have been optimized to achieve the most optimal outcome. The system collects live packet data through Scapy which uses the same analysis method as the training process to examine packet contents. The trained model uses the derived feature vector from analysis to make predictions. The web interface displays detection results through two indicators which show confidence level and type identification status. The project includes a chatbot system which informs users about observed attack types while providing simple solutions to maintain user engagement with the application.

VII. SYSTEM ARCHITECTURE

A system function architectural design requires four fundamental Design Elements for its architectural design. The Design Elements Include: Data Collection Layer: The layer enables users to enter network data through manual input while it also captures live traffic through automatic Scapy library-based traffic collection. The system serves as the main system for network stream data acquisition which users will process later. The Data Pre-processing and Features Layer: The layer removes unnecessary data packets while it extracts important data features which help create models that will identify network anomalies. The Machine Learning Detection Layer: The layer uses machine learning and deep learning models which include K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) to study preprocessed features and detect and classify and respond to networkanomalies. The system uses a recommendation module to show users which mitigation actions they should take. The system contains a chatbot which answers user questions while providing additional support. The system detects network anomalies with its capability to identify unusual network patterns in real time.

A. Architecture Workflow Explanation

The proposed application enables accurate network anomaly detection which operates almost in real-time detection mode. The system uses two methods for data acquisition which include manual entry of packet parameters and live traffic capture through the Scapy library. The system supports both controlled testing and real-time deployment through its two testing methods. The preprocessing stage transforms raw packet data through validation and encoding and normalization processes which produce structured feature vectors for analysis.

The detection module receives these features which use machine learning and deep learning models that include K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models. The models acquire knowledge of network traffic patterns which enables them to detect real-time traffic anomalies. The system generates an output after classification which includes the predicted label and the corresponding attack category and a confidence score that shows prediction reliability. The results get sent to the presentation layer which handles visualization and user interaction.

The LSTM model achieved the highest overall performance with an accuracy rate of 96% and the RNN model followed with 93% accuracy while the KNN model obtained 89% accuracy. The three models show high precision and recall values which demonstrate their ability to accurately identify both standard and unusual network traffic patterns. The models demonstrate high F1-scores which show their ability to maintain a balanced rate of false positives

and false negatives. LSTM successfully learns to identify extended time relationships in network flow data which enables it to detect intricate attack patterns that occur in sequences.

The system brings better network security monitoring efficiency through its dual capability of displaying security information in real time and generating advanced security reports. The web interface allows users to track detected anomalies together with attack types and prediction accuracy levels in a simplified analysis system. The chatbot-enabled recommendation system offers users relevant security advice together with detailed explanations of detected threats, which helps users identify attack patterns and defense strategies. The framework supports current cybersecurity needs because it combines precise detection capabilities with continuous system monitoring and accessible user systems in research.

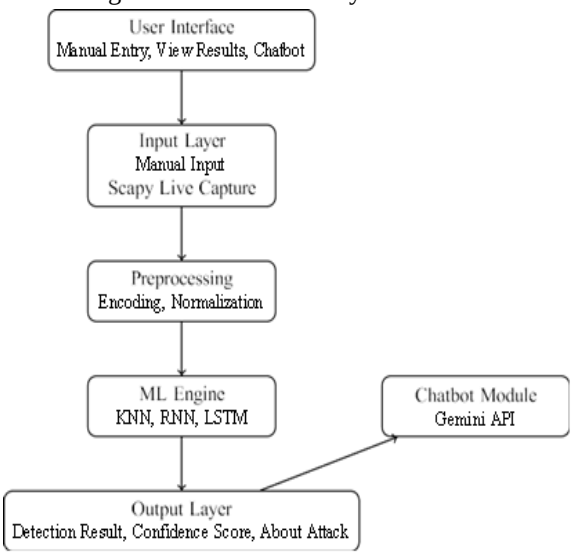

Fig. 1. System architecture of the proposed network anomaly detection system integrating manual input, Scapy live capture, ML-based classification, and chatbot support.

VIII. RESULTS AND DISCUSSION

The proposed Network Anomaly Detection System demonstrated its testing power through its evaluation across all testing conditions which were tested under equal testing conditions. The testing procedure required testing both offline data and Scapy real-time data within a controlled environment which restricted all testing activities. The testing dataset required division into training and testing datasets according to a specific ratio because it was essential to decrease potential overfitting risks.

The System's accuracy and precision and recall and F1 measurements were evaluated through testing both offline testing datasets and real-world Scapy system traffic. The above-mentioned measurements clearly identified and showcased the level of effectiveness of traffic anomaly events in real time, the number of FP that are produced, and also indicated system reliability. The System tests were conducted through various traffic patterns and attacks which will provide complete test results according to the testing requirements while the subsequent analysis will show how the system responds and display its anomalies through the evaluation of latency and response time and visualization degree.

The experiments revealed that LSTM performance exceeded all other tested models because it achieved a 96 percent accuracy rate, while RNN reached 93 percent and K-Nearest Neighbors attained 89 percent accuracy. The models achieved high precision and recall scores which enabled them to accurately detect regular and abnormal network traffic patterns. The models established their ability to balance false positive and false negative results through their respective F1-scores. The evaluated models showed that LSTM performed better than other options because it could recognize extended time patterns in network traffic, which helped identify intricate and developing attack methods. Analyses of the false-positive and false-negative rates were done along with analysis on the standard performance metrics.

The KNN system produced more false positive results because it relied on distance measurement to determine similarity among its features. The RNN system decreased false positive results when it learned traffic data patterns but it still made errors when detecting some types of attacks. The LSTM model provided the most accurate results between false positives and false negatives which made it ideal for real-time systems that required precise and trustworthy alert generation. The accuracy results from KNN RNN and LSTM models are shown in Fig 2 while Table I presents complete performance metrics which include accuracy and precision and recall and F1-score.

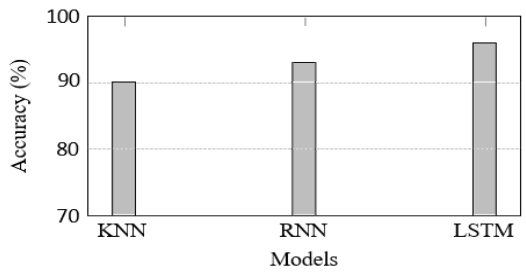

Fig. 2. Model Accuracy Comparison for KNN, RNN, and LSTM Used in the Anomaly Detection System

Table I. Model performance comparison using accuracy, precision, recall, and f1-score

Model	Accuracy	Precision	Recall	F1-Score
KNN	89%	0.88	0.87	0.87
RNN	93%	0.92	0.93	0.92
LSTM	96%	0.95	0.96	0.95

IX. CONCLUSION

The research showed that machine learning models could analyze complete network data when users entered data either through manual methods or during live operations. The proposed methods achieved the required accuracy performance together with a very small false positive detection rate while they provided recommendations to enhance the security protection of network data. The upcoming research will implement IoT sensors to enhance data processing capacity for live analysis while transitioning all operations to the cloud.

ACKNOWLEDGMENT

The authors want to thank Professor Harishma K V, Assistant Professor of the Department of Computer Science and Engineering, for providing them with the guidance, encouragement, and technical support they needed during the development of this work. The authors would like to extend their gratitude to the Department of Computer Science and Engineering for its support in providing the facilities complete this research.

REFERENCES

[1] N. N. Chandel, "Enhancing Cyber Security Through Machine Learning-Based Anomaly Detection," *International Journal of Engineering Research & Technology (IJERT)*, 2024.

[2] W. Marfo *et al*., "Network Anomaly Detection Using Federated Learning," *arXiv preprint*, 2023.

[3] Y.-C. Wang *et al*., "Network Anomaly Intrusion Detection Based on Deep Learning Approach," *Sensors*, vol. 23, no. 4, Art. no. 2171, 2023.

[4] R. Sharma *et al*., "Real-Time Intrusion Detection Using Machine Learning with Streaming Data Processing," in *Proc. IEEE Conference*, 2023.

[5] M. Azhar, S. Perveen, A. Iqbal, and B. Lee, "IDRandom-Forest: Advanced Random Forest for Real-Time Intrusion Detection," *IEEE Access*, vol. 12, pp. 3443408–3443421, 2024.

[6] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means + Random Forest and Deep Learning," *IEEE Access*, vol. 9, pp. 100–113, 2021.

[7] D. S. Kim, S. M. Lee, and J. S. Park, "Building Lightweight Intrusion Detection System Based on Random Forest," in *Advances in Neural Networks (ISNN), Lecture Notes in Computer Science*, vol. 3973, Springer, 2006.

[8] L. Yang, S. Gao, *et al*., "Griffin: Real-Time Network Intrusion Detection System via Ensemble of Autoencoder in SDN," 2022.

[9] I. Sharafaldin, A. H. Lashkari, and A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.

[10] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," *arXiv preprint*, 2019.

[11] A. Andalib and V. T. Vakili, "An Autonomous Intrusion Detection System Using an Ensemble of Advanced Learners," *arXiv preprint*, 2020.

[12] P.-F. Marteau, "Random Partitioning Forest for Point-Wise and Collective Anomaly Detection—Application to Intrusion Detection," *arXiv preprint*, 2020.

[13] H.-T. Pai, Y.-H. Kang, and W.-C. Chung, "Interpretable Generalization Mechanism for Detecting Anomaly and Identifying Network Intrusion Techniques," *arXiv preprint*, 2024.

[14] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Dataset for Network Intrusion

Detection Systems," in *Proc. Military Communications and Information Systems Conference (MilCIS)*, IEEE, 2015.

[15] R. Kumar and P. Singh, "Deep Neural Network-Based Intrusion Detection Using Flow-Level Features," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1123–1135, 2023.

[16] A. B. Hassan, M. Y. Idris, and K. Al-Harbi, "Adaptive Hybrid Machine Learning Model for Real-Time Network Anomaly Detection," *Computers & Security*, vol. 139, 2024.

[17] S. R. Patel and L. Kavitha, "Improving IDS Accuracy Using Optimized KNN and Autoencoder Fusion Model," *Journal of Network and Computer Applications*, vol. 215, 2023.

[18] J. Wu, Y. Tan, and H. Zhou, "Lightweight CNN-LSTM Architecture for High-Performance Intrusion Detection Systems," *Sensors*, vol. 22, no. 18, 2022.

[19] M. D. Sharma and A. Verma, "A Comparative Study of ML Algorithms for Detecting Cyber Threats in Enterprise Networks," *International Journal of Information Security Science*, vol. 13, no. 2, pp. 88–102, 2024.

[20] F. Ahmed and T. A. Khan, "Anomaly Detection Using Enhanced Feature Engineering and Gradient Boosting," *Applied Intelligence*, vol. 53, no. 7, pp. 9123–9139, 2023.

[21] G. Li, S. Zhou, and R. Wei, "Federated Deep Autoencoders for Privacy-Preserving Intrusion Detection," *Future Generation Computer Systems*, vol. 152, pp. 590–602, 2024.

[22] P. K. Rout, N. Das, and S. Mishra, "Hybrid IDS Using Random Forest and Bi-Directional LSTM for Cloud Environments," *Journal of Cloud Computing*, vol. 12, no. 1, 2023.

[23] D. K. Lee and C. Park, "Self-Supervised Learning for Network Traffic Anomaly Detection," *arXiv preprint*, 2023.

[24] Y. Sun, L. Chen, and J. Zhao, "Transformers-Based Network Intrusion Detection System with Multi-Head Attention," *IEEE Access*, vol. 12, pp. 156320–156332, 2024.

[25] A. Gupta and R. Chatterjee, "High-Speed Network IDS Using Reinforcement Learning and Deep Q-Networks," *Expert Systems with Applications*, vol. 238, 2024.