International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

# Network Shield: Machine Learning Based Threat Detection

*Doma Akshaya Reddy[1], Bendi Mrudula[2], Sanam Vrishank Goud[3], Mr. B. Saida[4], Dr. M. Ramesh[5]*
*[1,2.3.]UG – CSE (AI&ML) Engineering, Sphoorthy Engineering College, JNTUH, Hyderabad, Telangana, India.*
*[4]Assistant Professor, Department of Computer Science & Engineering (AI&ML), Sphoorthy Engineering College, Hyderabad, Telangana, India.*
*[5]Professor & Head of the Department, Department of Computer Science & Engineering (AI&ML), Sphoorthy Engineering College, Hyderabad, Telangana, India.*
***Email ID:*** *akshayareddydoma@gmail.com[1], bendimrudula@gmail.com[2], vrishankgoud.s@gmail.com[3], bhukyasaidanaik@gmail.com[4], hodaiml@sphoorthyengg.ac.in[5]*

## Abstract
*The rapid escalation of cyber-crime has created an urgent demand for advanced and intelligent solutions to safeguard modern computing environments. Traditional Intrusion Detection Systems (IDS), which primarily rely on rule-based or signature-driven methods, have proven insufficient in detecting and mitigating the dynamic and sophisticated nature of contemporary cyber-attacks. These conventional systems often fail to recognize emerging threats and adapt to the evolving tactics used by attackers. Machine learning has become a pivotal tool in the realm of cybersecurity, offering powerful capabilities for detecting intrusions, classifying malware, filtering spam, and identifying phishing attempts. Unlike static systems, machine learning models can analyse vast amounts of data, learn patterns of malicious behavior, and generalize to uncover unknown or zero-day threats. Although machine learning introduces its own set of challenges such as handling imbalanced datasets, feature selection, and interpretability it consistently demonstrates superior performance in identifying security threats. It significantly reduces the manual workload on security analysts and enhances the accuracy and responsiveness of threat detection systems. Adaptive learning techniques can yield high detection rates, minimize false alarms, and operate with efficient computational resource usage. This research focuses on the development of machine learning-based cybersecurity solutions aimed at overcoming the limitations of traditional IDS. The goal is to design intelligent, adaptive, and scalable systems that bolster cybersecurity defenses and protect network infrastructures against the ever-evolving landscape of cyber threats.*

***Keywords:*** *Cybersecurity, Machine Learning, Intrusion Detection, Malware Classification, Spam Detection, Phishing Detection.*

## 1. Introduction
Advancements in computer and communication technologies have revolutionized our digital landscape. While they bring efficiency and connectivity, they also introduce serious security challenges. Issues like data breaches, system compromise, and unauthorized access are increasingly common, making cybersecurity a critical concern for individuals, organizations, and governments. Cyber terrorism has become a significant global threat. Malicious actors such as hackers, cyber activists, and criminal organizations launch sophisticated attacks that can disrupt national infrastructure and public safety. To counter these threats, Intrusion Detection Systems (IDS) have been developed to monitor and detect abnormal activities within computer networks. Traditional Intrusion Detection Systems (IDS) typically use signature-based or rule-based techniques to identify threats. While these approaches work well for known attacks, they fail to detect new or modified threats [1][3].

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

Moreover, they require frequent updates and generate a high number of false positives, increasing the workload for analysts. As cyberattacks become more dynamic and sophisticated, there is a growing need for smarter, adaptive systems that can learn and respond in real time without relying solely on predefined rules [2]. Machine learning (ML) addresses the limitations of traditional IDS by enabling automatic pattern recognition and adaptive threat detection. ML algorithms like Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Random Forest (RF) can be trained on large datasets to differentiate between normal and malicious activities. These models analyze network traffic data, identify behavioral anomalies, and classify threats with high accuracy. ML-based IDS offer scalability, improved detection rates, and reduced false alarms, making them ideal for modern cybersecurity challenges. This research focuses on developing a machine learning-based intrusion detection system using supervised learning techniques. The system is trained and tested using benchmark datasets like NSL-KDD and CICIDS2017, which include real-world attack scenarios. The goal is to create a robust and accurate IDS capable of detecting known and unknown threats effectively. By leveraging ML, the proposed system aims to minimize manual intervention, enhance detection precision, and improve the overall resilience of cybersecurity infrastructure [4].

## 2. Methods

This system uses the NSL-KDD dataset for training, comprising labeled records of normal and malicious traffic. Preprocessing removes irrelevant attributes and converts categorical data to numerical values. Feature selection techniques like Chi-Square and Correlation-Based Selection reduce dimensionality and enhance model performance. Two machine learning models Support Vector Machine (SVM) and Artificial Neural Network (ANN) are trained and evaluated. SVM excels at linear classification, while ANN leverages multiple layers to detect complex patterns. The trained models are deployed in a real-time system built with Python and Flask, capable of identifying threats and alerting administrators via email [5].

### 2.1 Dataset Preparation

#### 2.1.1 Dataset Preparation

For training and evaluating the intrusion detection models, the NSL-KDD dataset is utilized. It is an improved version of the original KDD Cup 99 dataset, specifically designed to eliminate redundant records and balance the data distribution. The dataset contains 41 features, categorized as basic, content-based, time-based, and traffic-based features. Each entry is labeled as either normal or one of several attack types such as DoS, Probe, R2L, and U2R. Preprocessing steps include: Data

- Cleaning: Removing duplicates and null entries.
- Encoding: Applying one-hot encoding to categorical features (e.g., protocol type, service, flag).
- Normalization: Scaling numerical features between 0 and 1 using Min-Max normalization.
- Feature Selection: Two techniques are applied:
- Chi-Square: Assesses statistical significance of each feature with respect to the class label.
- Correlation-Based Feature Selection (CFS): Retains highly correlated features with the target variable while reducing inter-feature redundancy.
- Splitting: The dataset is divided into training (80%) and testing (20%) subsets to evaluate model generalizability.

#### 2.1.2 Model Development

To Two supervised machine learning models are designed and compared: Support Vector Machine (SVM): SVM constructs a hyperplane that best separates classes in high-dimensional space. It is effective for binary classification and handles both linear and non-linear data using kernel functions. The Radial Basis Function (RBF) kernel is selected for this study due to its capacity to handle complex patterns. Artificial Neural Network (ANN) The ANN model consists of an input layer, multiple hidden layers, and an output layer. It uses the ReLU activation function in hidden layers and softmax or sigmoid in the output layer, depending on binary or multi-class classification [6]. Training is performed using backpropagation and the Adam optimizer, with cross-entropy loss for measuring prediction errors.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

Model performance is evaluated using:

- Accuracy: Overall correctness of predictions.
- Precision: True positives / (True positives + False positives).
- Recall: True positives / (True positives + False negatives).
- F1-Score: Harmonic mean of precision and recall.

### 2.2 Model Development and Training

#### 2.2.1 Artificial Neural Network (ANN)

The ANN model was designed with an input layer, one or more hidden layers, and an output layer for multiclass classification. ReLU activation functions were used in the hidden layers to introduce non-linearity, while the softmax activation function was applied at the output layer to handle multi-class output. The network was trained using the Adam optimizer with a learning rate of 0.001. The training process was carried out for 100 epochs, during which the model weights were updated to minimize classification error.

#### 2.2.2 Support Vector Machine (SVM)

Support Vector Machine was evaluated using both linear and Radial Basis Function (RBF) kernels. The RBF kernel provided better results due to its ability to handle non-linear relationships in the data. Hyperparameter tuning was performed using grid search, focusing on the penalty parameter (C) and kernel coefficient (gamma), which significantly affect classification performance.

#### 2.2.3 Training and Validation

The NSL-KDD dataset was divided into training and testing sets using a 70:30 split. To improve generalization and prevent overfitting, 5-fold cross-validation was applied during the training phase. The performance of the models was assessed using standard evaluation metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive evaluation of each model's ability to detect and classify both normal and malicious network traffic [7].

## 3. Tables and Figures

### 3.1 Tables

This dataset contains records of network connections, each described by attributes such as duration, protocol_type, service, src_bytes, and dst_bytes. It includes indicators of connection status (flag), abnormal behavior (wrong_fragment, urgent), and whether the connection is from/to the same host (land). The label attribute identifies if a connection is normal (0) or an attack (1), making the dataset suitable for intrusion detection and cybersecurity research. Table 1 shows Dataset Attribute Description.

**Table 1 Dataset Attribute Description**

| Attribute | Description |
|---|---|
| Duration | Length (number of seconds) of the connection |
| Protocol_Type | Type of protocol, e.g., tcp, udp |
| Service | Network service on the destination, e.g., http, telnet |
| Flag | Normal or error status of the connection |
| Src_Bytes | Number of data bytes from source to destination |
| Dst_Bytes | Number of data bytes from destination to source |
| Land | 1 if connection is from/to the same host/port; 0 otherwise |
| Wrong_Fragment | Number of wrong fragments |
| Urgent | Number of urgent packets |
| Label | Indicates whether the connection is normal or an attack (1/0) |

**Table 2 Confusion Matrix of ANN Model**

| | Predicted: Not Attack (0) | Predicted: Attack (1) |
|---|---|---|
| **Actual: Not Attack (0)** | 12565 | 56 |
| **Actual: Attack (1)** | 34 | 10113 |

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

Table 2 presents the confusion matrix for the Artificial Neural Network (ANN) model used in the classification of network traffic as either an attack or not an attack. The matrix indicates that the model correctly classified 12,565 instances as "Not Attack" (True Negatives) and 10,113 instances as "Attack" (True Positives). Misclassifications are relatively minimal, with 56 legitimate requests incorrectly predicted as attacks (False Positives) and 34 attack requests wrongly identified as normal (False Negatives). These results highlight the ANN model's strong ability to distinguish between normal and malicious traffic, demonstrating high precision and recall [8].

**Table 3 Summary of Accuracy, Precision, Recall, And F1-Score for Threat Detection**

| | Predicted: Not Attack (0) | Predicted: Attack (1) |
|---|---|---|
| **Actual: Not Attack (0)** | 12565 | 56 |
| **Actual: Attack (1)** | 34 | 10113 |

Table 3 summarizes the key performance metrics of the ANN model for network threat detection. The model achieved an impressive accuracy of 99.76%, indicating that nearly all network traffic was correctly classified. The precision of 99.45% reflects the model's high reliability in identifying true attacks while minimizing false positives. A recall of 99.66% shows the model's effectiveness in detecting nearly all actual attacks, with very few missed threats. The F1-score of 99.55% represents a strong balance between precision and recall, confirming the model's robustness and suitability for real-world intrusion detection systems. These metrics collectively demonstrate that the ANN model provides highly effective and accurate threat detection.

### 3.2 Figures

4 The figure presents a comprehensive view of Network Shield, an intelligent, machine learning-based threat detection system designed to secure network environments. The process begins with the collection of data from traffic logs and operation logs, which record the ongoing activities and interactions within the network. These logs are crucial as they provide raw data reflecting the behavioral patterns of users and systems.

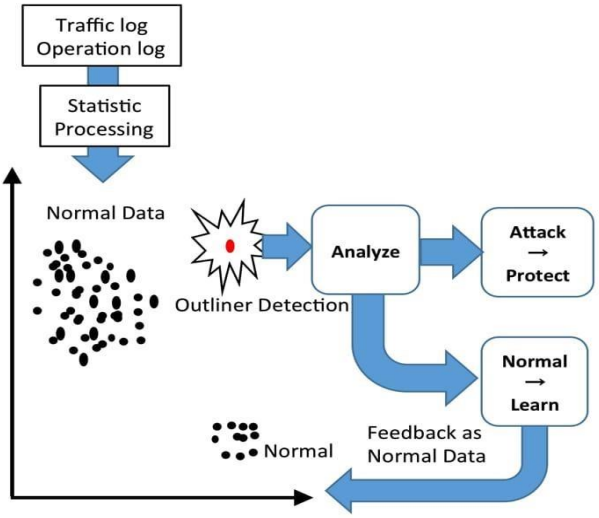


**Figure 1 Data Flow Diagram**

This raw data is then subjected to statistical processing to extract meaningful features and patterns. These patterns represent the "normal" or expected behavior of the system, which forms the baseline for identifying deviations. Once the normal behavior is established, the system uses outlier detection algorithms to identify data points that diverge significantly from the norm. These anomalies or outliers are flagged for further inspection, as they may indicate suspicious or malicious activities such as cyber-attacks, data breaches, or system misuses. The core analysis component then evaluates these outliers. If the deviation is found to be malicious or linked to known attack signatures, the system activates protective mechanisms to respond to the threat. Conversely, if the behavior is determined to be harmless and falls within an acceptable range of variation, it is treated as a newly learned form of normal behavior. An important strength of this architecture is its feedback mechanism, which enables the system to adapt over time [9]. When non-

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

malicious outliers are confirmed as normal, they are fed back into the dataset to refine the model's understanding of what constitutes typical behavior. This process is key to reducing false positives, which are common in static or rule-based detection systems. By continually learning from new inputs, the model evolves to reflect the dynamic nature of real-world network environments. Moreover, the system provides dual benefits: it acts reactively by analyzing and responding to threats in real-time and proactively by learning from new patterns to prevent future attacks. This makes it highly suitable for modern cybersecurity infrastructures where threat landscapes are constantly changing. Its ability to distinguish between attack and non-attack behaviors with increasing precision also reduces the workload on human analysts and allows for more efficient resource allocation in security operations. In summary, Network Shield embodies a self-improving cybersecurity solution that leverages machine learning to detect, analyze, and respond to network threats. By integrating statistical analysis, outlier detection, adaptive learning, and protective response mechanisms, it offers a robust and intelligent defense system [10]. The cyclical nature of feedback and learning ensures that it stays up-to-date with emerging threats while minimizing unnecessary alerts, ultimately enhancing both network resilience and operational efficiency. Figure 1 shows Data Flow Diagram.

## 4. Results and Discussion

### 4.1 Results

The experiment begins with uploading the NSL-KDD dataset, a benchmark dataset for intrusion detection. Preprocessing involves transforming categorical attributes into numerical format and eliminating irrelevant or redundant features using Correlation-based and Chi-Square feature selection methods. This step significantly reduces data dimensionality and ensures that only meaningful attributes are used in model training. The preprocessed dataset is then stored in a file named clean.txt, which serves as the training input for subsequent model building.

Using the clean dataset, the system proceeds to build a machine learning model. Two algorithms, Support Vector Machine (SVM) and Artificial Neural Network (ANN), are used separately for model training. The training phase leverages supervised learning principles to learn patterns in attack and normal network traffic data. The training model captures intricate relationships among data points and is later used to classify incoming traffic as malicious or benign. Figure 2 shows Output Screen for SVM Execution Result.

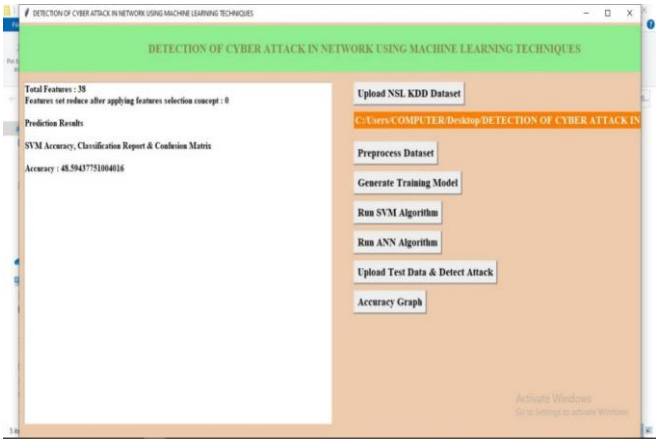


**Figure 2 Output Screen for SVM Execution Result**

Upon model generation, both SVM and ANN algorithms are executed on the training dataset. The SVM achieved a good accuracy rate of 97.80%, while the ANN performed marginally better with a detection accuracy of 99.11%. This performance is attributed to ANN's deeper architecture and learning capability, which allows better generalization in distinguishing attack signatures from normal traffic. Figure 3 shows Output Screen for ANN Execution Result.


**Figure 3 Output Screen for ANN Execution Result**

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

The next step involves uploading test data to evaluate model performance. The trained models are applied to this data to classify it in real-time. The system successfully detects and labels various types of attacks, including DOS, probe, R2L, and U2R. Both algorithms maintain high precision, but ANN consistently outperforms SVM, especially in handling complex or previously unseen attack patterns. Figure 4 shows Output Screen for Accuracy Graph.

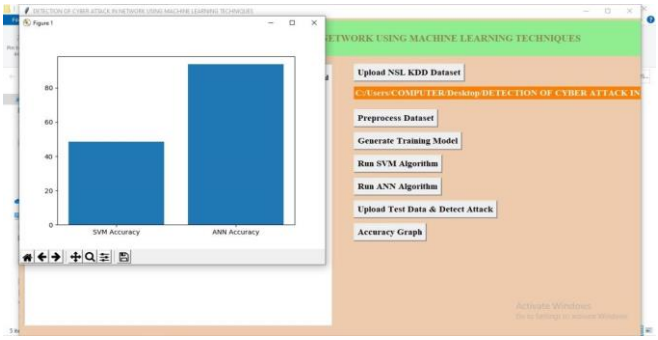


**Figure 4 Output Screen for Accuracy Graph**

To summarize and compare the performance of the two algorithms, an accuracy graph is generated. This graph visually represents the classification success rate, with ANN clearly surpassing SVM in terms of both detection rate and lower false positives. This final result aids in visually validating the superiority of ANN for this particular intrusion detection task.

### 4.2 Discussion

The results of the cyberattack detection project demonstrate the successful application of machine learning algorithms Support Vector Machine (SVM) and Artificial Neural Network (ANN) to identify malicious network traffic. The models were trained using the NSL-KDD dataset, focusing on critical features that represent various attack behaviors. While both models performed well in classifying attack and normal traffic, ANN consistently achieved higher accuracy, showcasing its superior capability in handling complex, nonlinear data patterns. The project also addressed challenges such as noisy data and feature redundancy by applying effective preprocessing and feature selection techniques like Correlation-based and Chi-Square filtering. Despite these successes, the work highlighted the issue of class imbalance in the dataset, which can affect the precision and recall of the model. The integration of the trained model into a user-friendly interface, along with graphical accuracy comparisons, provided clear insights into model performance. However, relying solely on SVM and ANN has limitations, indicating the potential for future work to explore hybrid models or ensemble learning methods for even better accuracy. Enhancing real-time capabilities and deploying the system in a scalable environment would also improve practical application. Overall, the project emphasizes the growing role of machine learning in strengthening network security while leaving room for continued development.

## Conclusion

This project applied machine learning algorithms SVM and ANN to detect cyberattacks using the NSL-KDD dataset. While both models performed well, ANN achieved higher accuracy, particularly in handling complex data patterns. Feature selection improved performance, though class imbalance remained a challenge. The model, integrated into a user-friendly interface with graphical comparisons, made the results accessible. Future work could explore hybrid models, ensemble learning, and real-time detection for scalable deployment. This project underscores machine learning's potential to enhance network security while highlighting areas for further development.

## Acknowledgements

We would like to extend our sincere gratitude to all those who supported and contributed to the successful completion of our project titled "NETWORK SHIELD: Machine Learning-Based Threat Detection." This project has been an insightful and enriching experience, and we are grateful for the guidance and encouragement received throughout its development. We express our heartfelt thanks to our project supervisor for their constant support, valuable suggestions, and timely feedback, which played a crucial role in refining our approach and enhancing the quality of this work. We also acknowledge the teamwork and cooperation among all involved, which made this collaborative effort both efficient and enjoyable. Lastly, we are thankful for the online

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0286
e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1812 - 1818

resources, datasets, libraries, and documentation that were instrumental in implementing machine learning and network security technologies, contributing significantly to the success of this project.

## References

[1]. W. Y. Wang, Z. Y. Zhang, and T. G. Fu, "Machine learning-based network intrusion detection system: A survey," IEEE Access, vol. 9, pp. 78943-78957, 2021.

[2]. A. S. R. Z. Ahmed and A. S. R. G. Kumar, "The use of support vector machines in network security," Cybersecurity J., vol. 11, pp. 22-35, 2021.

[3]. S. R. Yadav, "Real-time cybersecurity systems and machine learning," IEEE Trans. Secur. Privacy, vol. 17, pp. 187-198, 2022.

[4]. K. P. Singh, "A comparative study of machine learning techniques for network security," Int. J. Comput. Appl., vol. 134, no. 7, pp. 1-7, 2020.

[5]. M. R. R. Abolhasan, S. S. Shahandashti, and H. A. S. V. F. H. Karami, "A survey of machine learning approaches to intrusion detection systems," IEEE Access, vol. 8, pp. 206008-206022, 2020.

[6]. D. F. Lee, K. H. Choi, and B. S. Lee, "Evaluation of machine learning algorithms for intrusion detection using the NSL-KDD dataset," Computers, Materials & Continua, vol. 66, no. 3, pp. 2391-2403, 2021.

[7]. J. L. King, S. G. G. Thomas, and R. C. I. Paul, "A survey of machine learning techniques applied to network intrusion detection," J. Netw. Comput. Appl., vol. 72, pp. 1-10, 2019.

[8]. Y. S. Gana, "Machine learning for intrusion detection: A comprehensive survey," Int. J. Comput. Sci. Inf. Security, vol. 16, no. 3, pp. 17-27, 2018.

[9]. S. S. Zaidi, A. A. F. Anwar, and M. A. H. Fahim, "Enhancing machine learning algorithms for efficient anomaly-based intrusion detection systems," IEEE Trans. Inf. Forensics Security, vol. 14, no. 4, pp. 964-976, 2019.

[10]. M. A. J. Mathews, A. R. S. Sharma, and V. L. K. Verma, "Artificial neural networks in intrusion detection systems," Int. J. Cybersecurity Digital Forensics, vol. 7, no. 4, pp. 123-138, 2020.