

# Leveraging Machine Learning for Proactive Network Security Threat Detection: Techniques, Challenges, and Future Directions

Ahmed Mohammed Abdukareem<sup>1</sup>, Ahmed Zeyad Almhanna<sup>2</sup>

<sup>1</sup>Department of computer engineering techniques, University of Alnisour, Baghdad, Iraq

<sup>21</sup>Department of computer engineering techniques, University of Dijlah, Baghdad, Iraq

---

## Article Info

### Article history:

Received July, 7, 2025

Revised July, 30, 2025

Accepted Aug., 15, 2025

---

### Keywords:

Machine Learning  
Network Security  
Intrusion Detection  
Deep Learning  
Anomaly Detection

---

## ABSTRACT

The escalating sophistication and volume of cyber threats necessitate a paradigm shift from traditional, reactive network security measures towards proactive, intelligent detection systems. This paper investigates the application of Machine Learning (ML) techniques for enhancing network security threat detection. The core research problem addressed is the inadequacy of conventional signature-based and rule-based systems in identifying novel, zero-day, and polymorphic attacks effectively. We explore the fundamental ML paradigms – supervised, unsupervised, and reinforcement learning – detailing their applicability to various network security tasks such as intrusion detection, malware analysis, and anomaly identification. Key aspects of the ML pipeline, including feature selection, data preprocessing, and robust model evaluation metrics, are discussed. The paper reviews significant implementations and case studies, highlighting the performance of different ML algorithms using benchmark datasets like NSL-KDD, CIC-IDS, and UNSW-NB15. Despite promising results demonstrating ML's capability to improve detection accuracy and reduce false alarms, significant challenges remain. These include the high dimensionality of network data, the need for large-scale labeled datasets, the persistent issue of false positives/negatives, vulnerability to adversarial attacks, data privacy concerns, computational overhead, and the inherent difficulty in interpreting complex models (explainability). Future directions point towards the development of explainable AI (XAI), federated learning for privacy preservation, advanced reinforcement learning for autonomous response, hybrid modeling approaches, and strategies to counter adversarial manipulations. This research concludes that while ML offers powerful tools for bolstering network defenses, continuous research and development are crucial to overcome existing limitations and stay ahead of the evolving threat landscape.

---

## Corresponding Author:

Ahmed Mohammed Abdukareem  
Department of Computer Engineering Techniques, University of Alnisour  
fallujah, Anbar, Iraq  
Email: ahmed.abd.com@nuc.edu.iq

---

## 1. INTRODUCTION

The digital transformation has interconnected global systems, creating unprecedented opportunities but simultaneously expanding the attack surface for malicious actors. Network security, the practice of preventing and detecting unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources, is paramount for protecting critical infrastructure, sensitive data, and organizational reputation [1]. Traditional community security mechanisms, which includes firewalls, signed-primarily based infiltration gadget (IDS) and antivirus software program, have shaped the basis for cyber protection for many years. However, they

depend on the specially predefined rules and signature of the well-known chance, which quickly useless attacks, and by accident attacking beside the point, and zero. Threatening (apts)[2].The dynamic nature of the natural extent and modern-day threats of network site visitors overwhelming guide analysis and stable defense strategies.

This evolving danger landscape necessitates more wise, adaptive, and automated safety solutions. Machine Learning (ML), a subset of Artificial Intelligence (AI) that enables systems to research from statistics and improve overall performance over the years without being explicitly programmed, has emerged as a transformative generation in cybersecurity [3]. By analyzing enormous amounts of community records (e.G., packet headers, go with the flow statistics, logs), ML algorithms can perceive diffused styles, anomalies, and correlations indicative of malicious activity that frequently prevent traditional strategies. ML fashions can examine normal community conduct and flag deviations, allowing the detection of novel threats and improving the accuracy and efficiency of safety operations.

The significance of ML in network safety lies in its functionality to shift the safety posture from reactive to proactive. Instead of looking beforehand to a appeared signature, ML fashions can expect ability threats primarily based on decided styles, allowing quicker reaction times and mitigating functionality harm. This capability is vital in immoderate-stakes environments in which protection breaches ought to have catastrophic consequences. This research paper dreams to offer a comprehensive assessment of the software of Machine Learning techniques for network safety danger detection. The primary objectives are:

1. To review traditional network security methods and articulate their limitations in the context of modern cyber threats.
2. To explore the fundamental Machine Learning paradigms (supervised, unsupervised, reinforcement learning) and their specific applications in network threat detection.
3. To discuss critical components of the ML workflow, including data preprocessing, feature engineering, and model evaluation tailored for security contexts.
4. To examine real-world implementations, case studies, and benchmark datasets used in ML-based network security research.
5. To critically analyze the inherent challenges and limitations associated with deploying ML for threat detection, such as adversarial attacks and explainability issues.
6. To identify promising future research directions and technological advancements in the field.

This paper is structured as follows: Section 2 provides background on traditional security methods and reviews related work in ML for cybersecurity. Section 3 delves into specific ML techniques applicable to threat detection. Section 4 presents implementation details and case studies. Section 5 discusses the challenges and limitations. Section 6 explores future directions. Finally, Section 7 concludes the paper with a summary of key insights.

## **2. Background & Related Work**

### **2.1 Traditional Network Security Methods**

Conventional network security infrastructure typically comprises several layers of defense:

- **Firewalls:** These act as gatekeepers, controlling the traffic of the network based on predetermined security rules (e.g., blocking specific ports or IP addresses). While essential, traditional firewalls lack the intelligence to inspect packet contents deeply or detect sophisticated application-layer attacks. Next-Generation Firewalls (NGFWs) incorporate more advanced features like deep packet inspection (DPI) and application awareness, but still primarily rely on rules and signatures.
- **Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS):** IDS monitor network or system activities for malicious activities or policy violations and produce reports, while IPS actively block detected threats [1]. Signature-based IDS/IPS compare network traffic against a database of known attack patterns (signatures). They are effective against known threats but fail to detect zero-day exploits or variations of known attacks. Anomaly-based IDS attempt to identify deviations from a baseline of normal behavior, but often suffer from high false positive rates.
- **Antivirus (AV) Software:** Focuses on detecting, preventing, and removing malicious software (malware) primarily through signature matching. Polymorphic and metamorphic malware, which constantly change their code, can easily evade traditional AV detection.

- Security Information and Event Management (SIEM) Systems: Aggregate log data from various network devices and systems, correlating events to identify potential security incidents. While powerful for centralized monitoring, SIEM systems often rely on predefined correlation rules and can be overwhelmed by the volume of data, leading to missed detections or alert fatigue.

## 2.2 Limitations of Traditional Methods

The primary limitations driving the need for ML-based approaches include:

- Inability to Detect Novel Threats: Signature and rule-based systems are inherently reactive; they cannot identify threats for which no signature or rule exists.
- Scalability Issues: The exponential growth in network traffic volume and speed challenges the processing capacity of traditional systems, especially those performing deep packet inspection.
- High False Positives/Negatives: Anomaly detection systems often generate numerous false alarms (false positives) by flagging benign deviations, while signature-based systems can miss slightly altered known attacks (false negatives).
- Static Nature: Rules and signatures require constant manual updates by security experts, a process that struggles to keep pace with the rapid evolution of attack techniques.
- Encrypted Traffic: Increasing use of encryption (e.g., SSL/TLS) hinders inspection capabilities of many traditional tools, although ML techniques can sometimes analyze encrypted traffic patterns [4].

## 2.3 Related Work: Machine Learning in Cybersecurity

The application of ML to cybersecurity is not new, with early work focusing on spam filtering and basic anomaly detection. However, recent advancements in ML algorithms, coupled with increased computational power and data availability, have spurred significant research interest in network security [3, 5].

Several surveys and review papers provide broad overviews of ML for cybersecurity. Buczak and Guven [3] offered an early comprehensive survey on ML techniques for intrusion detection. Ahmad et al. [5] reviewed ML algorithms specifically for anomaly-based intrusion detection in networks. Xin et al. [6] provided a survey focusing on ML and data mining methods for cybersecurity, covering intrusion detection, malware analysis, and spam detection. More recently, surveys have focused specifically on the application of deep learning techniques, comparing various approaches and datasets [14].

Specific areas where ML has been actively researched include:

- Intrusion Detection: Numerous studies have applied supervised algorithms like Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), and Neural Networks (NN) to classify network traffic as normal or intrusive, often using benchmark datasets like KDD Cup '99, NSL-KDD, UNSW-NB15, and CIC-IDS [2, 7]. Unsupervised methods, such as K-Means clustering and Autoencoders, have been explored for detecting anomalies and zero-day attacks [4, 8].
- Malware Detection: ML is used for both static analysis (examining code without execution) and dynamic analysis (observing behavior during execution) to classify files or network flows associated with malware [9]. Deep learning models, particularly Convolutional Neural Networks (CNNs) applied to malware visualizations and Recurrent Neural Networks (RNNs) for analyzing behavioral sequences, have shown promise.
- Phishing Detection: ML models analyze email headers, content, and URLs to identify phishing attempts with higher accuracy than traditional filters.
- Network Traffic Analysis: ML helps in classifying traffic types (e.g., identifying specific applications or protocols) and profiling network behavior for anomaly detection or Quality of Service (QoS) management.

While existing literature demonstrates the potential of ML, many studies focus on specific algorithms or datasets, often under controlled conditions. There remains a need for research addressing the practical challenges of deploying these systems in real-world, high-throughput network environments, dealing with concept drift, and ensuring model robustness against adversarial manipulations [10]. This paper aims to synthesize these diverse applications, focusing on network threat detection while highlighting practical considerations and future pathways.

## 3. Machine Learning Techniques for Threat Detection

ML algorithms applied to network security can be broadly categorized into supervised, unsupervised, and reinforcement learning. The choice of technique depends on the specific security task, data availability (especially labeled data), and desired outcome (e.g., classification of known threats vs. detection of novel anomalies).

### 3.1 Supervised Learning

Supervised learning algorithms learn a mapping function from input features to output labels based on a labeled dataset. In network security, this typically involves training models on data where instances are pre-classified as 'normal' or 'malicious' (or specific attack types).

- **Algorithms:**

- **Support Vector Machines (SVM):** Effective in high-dimensional spaces, finding an optimal hyperplane to separate different classes. Useful for classifying intrusions or malware types.
  - **Decision Trees (DT) and Random Forests (RF):** DTs create tree-like models of decisions. RFs build multiple DTs (an ensemble) and aggregate their predictions, improving robustness and accuracy, often performing well in intrusion detection tasks [7].
  - **Naive Bayes:** classifier depends on probability based on Bayes' theorem with strong (naive) independence assumptions. Simple and efficient, often used as a baseline.
  - **K-Nearest Neighbors (KNN):** Classifies instances according to the maturity class among their 'k' nearest neighbors in the point space. Simple but can be computationally expensive for large datasets.
  - **Neural Networks (NN) and Deep Learning (DL):** Multi-layered networks capable of learning complex patterns. Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs – useful for spatial hierarchies, e.g., in traffic visualization), and Recurrent Neural Networks (RNNs – suitable for sequential data like packet sequences or logs) are increasingly used [4, 9, 14].
- **Use Cases:** Classifying known types of attacks (e.g., DoS, Probe, U2R, R2L), identifying specific malware families, detecting phishing websites based on URL features.
  - **Requirements:** Requires large amounts of accurately labeled training data, which can be difficult and expensive to obtain in cybersecurity.

### 3.2 Unsupervised Learning

Unsupervised learning algorithms identify patterns and structures in unlabeled data. They are particularly valuable for discovering novel threats and anomalies that do not match known signatures.

- **Algorithms:**

- **Clustering (e.g., K-Means, DBSCAN):** Groups similar data points together. Outliers or small clusters may represent anomalous or malicious activity [8]. K-Means partitions data into 'k' predefined clusters, while DBSCAN identifies density-based clusters and outliers.
  - **Anomaly Detection (e.g., Isolation Forest, Local Outlier Factor (LOF), One-Class SVM):** Specifically designed to identify rare items or events that deviate significantly from the majority of the data. Isolation Forest efficiently isolates anomalies using random trees.
  - **Dimensionality Reduction (e.g., Analysis of the basic Component (PCA), t-SNE):** Reduces the number of features while preserving important information. Useful for visualization and improving the performance of other ML algorithms by mitigating the 'curse of dimensionality'. Autoencoders (a type of neural network) can also be used for dimensionality reduction and anomaly detection by learning compressed representations of normal data; high reconstruction errors indicate anomalies [4].
- **Use Cases:** Detecting zero-day attacks, identifying unusual network traffic patterns, discovering covert communication channels, user behavior analysis.
  - **Advantages:** Does not require labeled data, making it suitable for dynamic environments where new threats constantly emerge.

### 3.3 Reinforcement Learning (RL)

Reinforcement learning includes an agent learning optimal behaviors by the interaction with an environment and receiving rewards or penalties. While less mature in network security compared to supervised/unsupervised learning, RL holds potential for creating adaptive and autonomous defense systems.

- **Concepts:** Agent (e.g., firewall, IDS), Environment (network), State (current network conditions), Actions (e.g., block IP, allow traffic, throttle connection), Reward (e.g., positive for blocking attacks, negative for blocking legitimate traffic).
- **Algorithms:** Q-learning, Deep Q-Networks (DQN).
- **Potential Use Cases:** Autonomous incident response, dynamic firewall policy adaptation, optimizing honeypot interactions, automated penetration testing to find vulnerabilities. Surveys suggest growing interest in RL for designing autonomous cyber defense systems [15].
- **Challenges:** Defining appropriate reward functions, ensuring stability and safety, high sample complexity (requires many interactions).

### 3.4 Feature Selection and Engineering

The performance of any ML model heavily depends on the quality and relevance of the input features. Network data is often high-dimensional and complex.

- **Feature Sources:** Packet headers (IP addresses, ports, protocol flags, TTL), flow data (NetFlow, sFlow - summaries of conversations including duration, byte/packet counts), system logs, IDS alerts.
- **Feature Engineering:** Creating new features from existing ones (e.g., calculating traffic rates, connection duration statistics, ratios of packet types). Domain expertise is crucial here.
- **Feature Selection:** Choosing the most informative subset of features to reduce dimensionality, improve model performance, and decrease computational cost. Techniques include:
  - *Filter Methods:* Select features based on statistical properties (e.g., correlation, mutual information) independent of the ML algorithm.
  - *Wrapper Methods:* Use the performance of a specific ML algorithm to evaluate feature subsets. Computationally expensive.
  - *Embedded Methods:* Feature selection is integrated into the model training process (e.g., LASSO regression, feature importance in tree-based models).

### 3.5 Data Preprocessing

Raw network data is often unsuitable for direct use by ML algorithms and requires significant preprocessing:

- **Data Cleaning:** Handling missing values (imputation or removal), removing noise or irrelevant data.
- **Data Transformation:** Converting categorical features (e.g., protocol names) into numerical representations (e.g., one-hot encoding).
- **Normalization/Scaling:** Scaling numerical features to a common range (e.g., 0-1 or standard normal distribution) to prevent features with larger values from dominating the learning process.
- **Handling Imbalanced Data:** Security datasets are typically highly imbalanced, with malicious instances being rare compared to normal traffic. Techniques like oversampling minority classes (e.g., SMOTE - Synthetic Minority Over-sampling Technique), undersampling majority classes, or using cost-sensitive learning are necessary to prevent models from being biased towards the majority class [2].

### 3.6 Model Evaluation

Evaluating ML models for security requires metrics beyond simple accuracy, considering the costs associated with different types of errors.

- **Confusion Matrix:** A table summarizing classification performance (True Positives - TP, True Negatives - TN, False Positives - FP, False Negatives - FN).
- **Key Metrics:**
  - **Accuracy:**  $(TP + TN) / \text{Total}$ . Can be misleading in imbalanced datasets.
  - **Precision (Positive Predictive Value):**  $TP / (TP + FP)$ . Fraction of predicted positives that are actually positive. High precision minimizes false alarms.
  - **Recall (Sensitivity, True Positive Rate):**  $TP / (TP + FN)$ . Fraction of actual positives that are correctly identified. High recall minimizes missed detections.
  - **F1-Score:**  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$ . Harmonic mean of precision and recall, useful for balancing the two.
  - **False Positive Rate (FPR):**  $FP / (FP + TN)$ . Fraction of actual negatives incorrectly classified as positive. Crucial for operational usability (minimizing alert fatigue).
  - **ROC Curve (Receiver Operating Characteristic) and AUC (Area Under the Curve):** Plots TPR vs. FPR at various classification thresholds. AUC provides a single measure of overall performance across thresholds.
- **Validation Techniques:** Cross-validation (e.g., k-fold) is essential to ensure the model generalizes well to unseen data and is not overfitted to the training set.

## 4. Practical Implementation and Evaluation: DDoS Detection using Random Forest

To bridge the theory of ML techniques with practical application, this section details a concrete implementation focused on detecting Distributed Denial of Service (DDoS) attacks within network traffic, using the CIC-IDS2017 dataset (specifically, the Friday-WorkingHours-Afternoon-DDos.pcap\_ISCX.csv subset). The implementation utilizes Python with standard data science libraries (Pandas, Scikit-learn) and employs a Random Forest classifier.

## 4.1 Methodology

The implementation follows a standard ML pipeline:

1. **Data Loading and Preparation:** The CIC-IDS2017 dataset subset is loaded using Pandas. Column names are standardized. Crucially, the code handles data cleaning steps specific to this dataset, including replacing 'Infinity' strings and np.inf values with NaN. Binary labels are created where 'BENIGN' traffic is labeled as 0 and 'DDoS' traffic is labeled as 1. The code includes an option for balanced sampling to handle class imbalance if needed, although the final run used a specific sample size (50,000 instances). Rows containing NaN values after initial cleaning and potential sampling are dropped to ensure data integrity for the ML algorithms.
2. **Feature Selection and Preprocessing:** A key aspect of this specific implementation is the *intentional use of a very limited feature set*. Instead of using all available features, the pipeline focuses on a minimal subset, potentially to test model robustness or understand the predictive power of fundamental features. The selected numeric features are Fwd\_Header\_Length, Bwd\_Header\_Length, Average\_Packet\_Size, and Down/Up\_Ratio. Categorical features derived from TCP flags (e.g., FIN\_Flag\_Count, SYN\_Flag\_Count, etc.) are also identified and used if present and non-constant. A ColumnTransformer manages preprocessing:
  - o **Numeric Features:** Missing values (NaNs resulting from cleaning infinite values or inherent in the data) are imputed using the median strategy (CleanerTransformer, a modified SimpleImputer), followed by standardization using StandardScaler.
  - o **Categorical Features:** Missing values are imputed using the most frequent strategy, followed by OneHotEncoder to convert them into numerical format. Unknown categories encountered during testing are ignored. Any features not explicitly selected as numeric or categorical are dropped.
3. **Model Training:** A Random Forest classifier (RandomForestClassifier from Scikit-learn) is chosen as the model. To further explore performance under constraints or potentially improve generalization with limited features, the model is *heavily regularized* with restrictive hyperparameters:
  - o n\_estimators=50 (relatively few trees)
  - o max\_depth=3 (very shallow trees, limiting complexity)
  - o min\_samples\_leaf=100 (requiring a large number of samples in each terminal node)
  - o min\_samples\_split=200 (requiring a large number of samples to allow a split)
  - o class\_weight='balanced' (to mitigate potential class imbalance impact)The data is split into training (80%) and testing (20%) sets using stratification to maintain class proportions. The pipeline, including preprocessing and classification, is trained on the training set. Cross-validation (5-fold) is included in the code as a recommended practice to assess generalization before final training.
4. **Evaluation Metrics:** The model's performance on the unseen test set is evaluated using standard classification metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix, and the Receiver Operating Characteristic (ROC) curve with its Area Under the Curve (AUC).

## 4.2 Results and Discussion

The model trained with the severely limited feature set and restrictive hyperparameters yielded the following performance on the test set:

- Accuracy: 0.8985
- Precision: 0.8328
- Recall: 0.9972
- F1 Score: 0.9076

These metrics are visualized and further detailed through the following plots:

- ROC Curve:

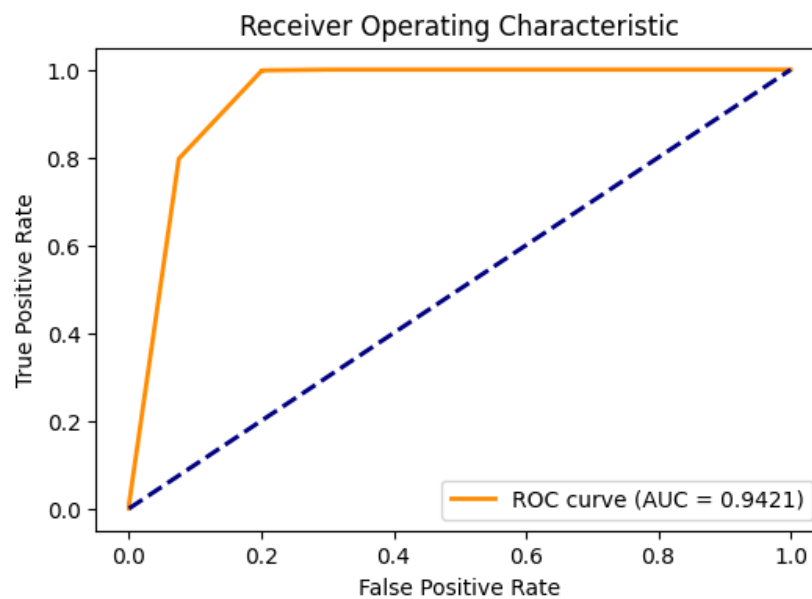


Figure 1: receiver operating Characteristic

- Confusion Matrix:

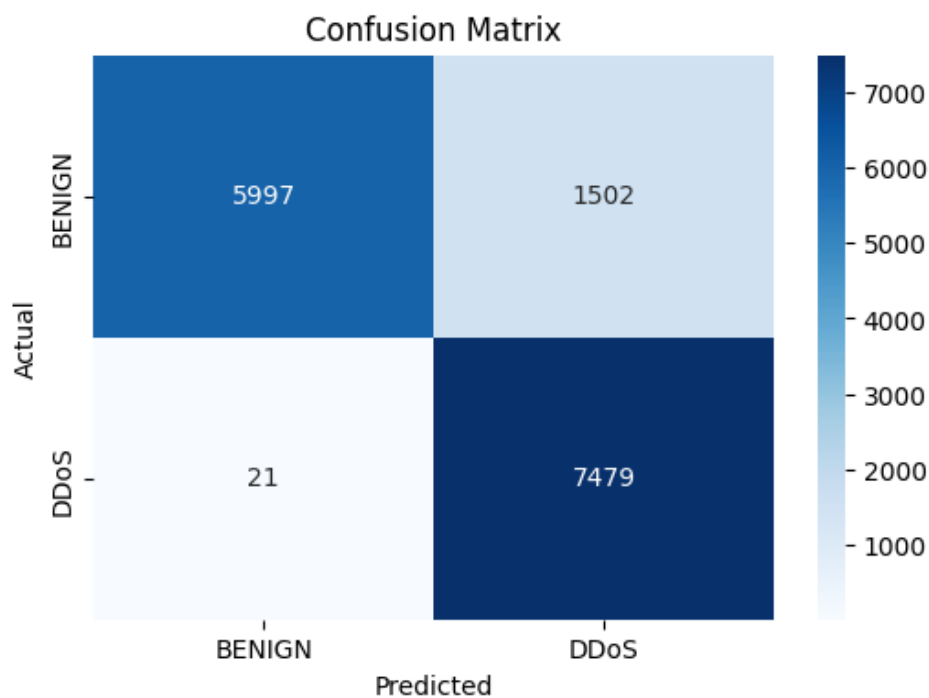


Figure 2: confusion matrix

## 5. Challenges & Limitation

Despite the promise, deploying ML for network security threat detection faces significant hurdles:

- **Data Quality and Availability:** Obtaining large, representative, and accurately labeled datasets for training supervised models is a major challenge. Real-world network data is often noisy, incomplete, and proprietary due to privacy concerns [6, 10]. Public datasets, while useful for research, may not reflect specific organizational environments or the latest threats. The creation of realistic datasets like Bot-IoT [16] helps, but the challenge remains.
- **False Positives and False Negatives:** Achieving a balance is critical. High false positives overwhelm security analysts (alert fatigue), leading to genuine threats being ignored. High false negatives mean attacks are missed. The acceptable threshold depends on the deployment context and the cost associated with each error type [3]. ML models, especially unsupervised ones, can struggle with high false positive rates.
- **Adversarial Machine Learning:** ML models themselves can be targets. Adversarial attacks involve crafting malicious inputs designed to fool the model:
  - *Evasion Attacks:* Modify malicious traffic slightly (e.g., padding packets, altering timing) to be misclassified as benign by the deployed model at detection time [10]. Seminal work has shown how even subtle input perturbations can fool deep learning models [11].
  - *Poisoning Attacks:* Inject manipulated data into the training set to compromise the learned model. Developing robust models resistant to such attacks is an active area of research.
- **Concept Drift:** Network behavior and attack techniques constantly evolve. A model trained on past data may become less effective over time as patterns change ('concept drift'). Continuous monitoring, periodic retraining, and adaptive learning mechanisms are necessary [5].
- **Interpretability and Explainability (XAI):** Many powerful ML models, especially deep learning, act as "black boxes," making it difficult to understand *why* they classify certain traffic as malicious. This lack of transparency hinders trust, debugging, regulatory compliance, and effective incident response [10]. Techniques like LIME (Local Interpretable Model-agnostic Explanations) [12] and SHAP (SHapley Additive exPlanations) are being developed but add complexity and may not fully capture global model behavior.
- **Computational Overhead:** Training complex deep learning models requires significant computational resources (CPU, GPU, memory) and time. Real-time detection on high-speed networks (e.g., 100 Gbps) demands highly optimized models and efficient inference pipelines [4].
- **Scalability:** Models must scale to handle the massive volume and velocity of traffic in modern networks without introducing significant latency.
- **Feature Engineering Complexity:** Designing effective features often requires deep domain expertise in both networking and security. Automating this process (e.g., via deep learning) is promising but challenging.
- **Data Privacy:** Using real network traffic for training improves important privacy considerations, potentially GDPR. Techniques such as anonymous data, discriminatory privacy and federated learning rules, are discovered to reduce these risks.

Tackling these challenges is critical for a success and reliable distribution of ML inside the operational network safety environment.

## 6. Future Directions

The area for network security is developing rapidly, with many promising roads for future research and development:

- **Explanation for Safety AI (XAI):** It is important to develop a naturally interpretable model or effective post-hoc explanation technique [12] It is important to enable trust and enable actionary insights in ML detection. Understanding why a notice was generated helps analysts validate the dangers and respond correctly [10].



- **Infated strength:** Research in creating ML models that are naturally flexible for theft and poison attacks, Paramount [11]. This provides unfavorable training (highlighting models for unfortunate examples during training), defensive distillation and certified defense that provides a proven guarantee against some attacks.
- **Federated Learning:** This privacy protection technique allows many organizations to work together to train a shared ML model without highlighting their raw network data [10]. Each organization trains the model locally, and only the model updates (eg gradients) are gathered, which arises according to the principles mentioned in the seminal work [13]. This data addresses privacy and enables trained models on more different data sets.
- **Learning reinforcement for an autonomous response:** RL provides the opportunity to create an autonomous system, move beyond detection that can not only detect the dangers, but can also take the right molding functions (eg, the overorganization of the firewall, distinguishes the compromised hosts) in real time, adapting to their strategies on the basis of observable results [3, 15].
- **Hybrid approach:** A combination of strength of different ML paradigms (eg, using unseen learning for informal identity after learning Supervision for classification) or integrating ML with traditional safety equipment (eg using ML to prioritize CEM warning or refine ID signature) can lead to more complex and effective solutions.
- **Transfer of learning and domain adaptation:** Techniques that allow trained models on a dataset or network environment can be adapted to the other quickly and effectively, which reduces the need for extensive marked data in new distribution.
- **Focus on specific domains:** IoT protection (to handle treatment equipment and unique protocols, Bot-OT [16]), Cloud Security (analysis of virtualized network traffic and log), and using dataset as Operational Technology (OT) Security, ML-technology.
- **Graph Neural Network (GNNs):** Using GNN for relationships between model network topology, communication patterns and institutions (hosts, users, files) provides a powerful way to detect complex, coordinated attacks such as APTS.
- **Automatic functional technique:** Technology such as raw network data (eg package catch) to automatically learn technology, which reduces the dependence on manual disability.
- **Continuous learning and adaptation:** Development systems are able to learn online and adapt to concept flow without the need for perfection, ensuring continuous performance in the dynamic environment.

Integration of ML with other AI regions, such as Natural Language Processing (NLP), will further strengthen the defensive abilities, to analyze the Danger Intelligence Report or Cyber Logic Systems.7.

## 7. Conclusion

8. Machine learning has proved undeniably emerged as a powerful and essential tool in the ongoing battle against refined network security threats. By taking advantage of the algorithm capable of learning complex patterns from the giant dataset, ML provides significant benefits on traditional signature-based and rules-based systems, especially in the detection of novels, zero-skin and polymorphic attacks. Monitored, insecure and reinforcement patterns provide unique abilities used on each different safety functions, from infiltration classification and detection of harmful software to identifying deviations and potential autonomous response.
9. This article has provided a comprehensive observation of the use of ML in network threat detection, discussed basic techniques, important implementation stages such as functional technique and pre -preparation, and model assessment strategies prepared for security domains. The reference reflects the ML opportunity to improve the comparison speeds and perform comparison of performance based on the reference dataset and to reduce false alarms.
10. However, practical distribution of ML in network security is not without significant challenges. Around the quality and availability of data, trade between false positivity and

negative, important vulnerability to unfavorable attacks, model lecturers (XAI), computational overhead, scalability and concept needs should be really effective and addressed in trusting.



11. Future research guidelines focus on dealing with these limitations. In XAI, unfavorable strength, federated learning, reinforcement learning and hybrid modeling methods huge promises. Innovation innovation is necessary to develop ML interest security systems that are not only accurate and effective, but also strong, adaptation, adaptation and relevant.
12. Finally, the machine represents a significant development in learning network security, transfer to focus Active, data -handled defense. While challenges remain, the progress of ml ices and is In a unique context of function, along with a deep understanding of his application Cyber security, more flexible and intelligent network paves the way to save Ever developed hazard landscape.

### 13. References

- [1] C. P. Pfleeger, S. L. Pfleeger, and J. Margulies, *Security in Computing*, 5th ed. Prentice Hall, 2015.
- [2] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Military Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
- [3] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, Secondquarter 2016.
- [4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
- [5] I. Ahmad, M. Bashari, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.
- [6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
- [7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, 2018, pp. 108–116.
- [8] L. D. S. Brenton, C. Leckie, M. Z. A. Bhuyan, M. M. Hassan and R. Ranjan, "Unsupervised Deep Learning for Network Intrusion Detection," *arXiv preprint arXiv:1803.05769*, 2018. [Online]. Available: <https://arxiv.org/abs/1803.05769>
- [9] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A Survey of Deep Learning Methods for Cyber Security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019.
- [10] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "Towards the Science of Security and Privacy in Machine Learning," *arXiv preprint arXiv:1611.03814*, 2016. [Online]. Available: <https://arxiv.org/abs/1611.03814>
- [11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014. [Online]. Available: <https://arxiv.org/abs/1412.6572>
- [12] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining (KDD)*, 2016, pp. 1135–1144.
- [13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, PMLR 54:1273-1282, 2017. [Online]. Available: <http://proceedings.mlr.press/v54/mcmahan17a.html>
- [14] M. A. Ferrag, L. Maglaras, S. Moschogiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2020.
- [15] M. Al-Hawawreh, K. A. Z. Razak, and A. S. Mohammad, "Reinforcement Learning-based Autonomous Cyber Defense Systems: A Survey," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–38, Jan. 2023.
- [16] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet

dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov. 2019.

## BIOGRAPHIES OF AUTHORS

	<b>Asst. Lecturer Ahmed Mohammed Abdulkarem</b> , Received His Msc. degree in the computer science from University of Anbar – Iraq in 2011 and Master Degree in computer science from Universiti Imam Riza international university Iran - Mashhad -2020. He has been a full-time lecturer in Alnisour university since 2024, Baghdad, Iraq. Ongoing He can be contacted at email: ahmed.abd.com@nuc.edu.iq.
	<b>Asst. Lecturer ahmed zeyad almhanna</b> Received His Msc. degree in the computer science from University of Baghdad – Iraq in 2017 and Master Degree in computer science from Universiti Imam Riza international university Iran - Mashhad -2020. He has been a full-time lecturer in Dijlah university since 2023, Baghdad, Iraq. Ongoing He can be contacted at email: ahmed.ziad@duc.edu.iq.