eISSN: Applied For

MERIT

Modern Explorations in Research, Innovation and Transformation

An International Peer-Reviewed Open Access Journal

Vol. 01 | Issue 01 | April 2026 | Pages xx–xx | Manuscript ID: MERIT-2026-0020

A COMPREHENSIVE SURVEY OF MACHINE LEARNING-BASED INTRUSION DETECTION FOR CYBERSECURITY THREAT CLASSIFICATION

Soumen Pore

Student, ECE Dept, Gargi Memorial Institute of Technology, MAKAUT, WB, India.
soumenpore7777@gmail.com

ABSTRACT

A large number of devices connected to the Internet, cloud technology, and digital communication methods have contributed to a greater incidence of risk events in the field of digital security. Existing detection methods using fingerprints, signatures, and rules cannot always necessarily identify so-called zero-day attacks and the new methods of penetration of businesses. Therefore, the need for ML-based detection emerges because of the new possibilities for detection of hidden deviations in the behavior of networks. The aim of this article is to provide a comprehensive overview of the existing ML-based intrusion detection systems (IDS). In the course of the work the analysis of standard chains of operations, assessment of the theory behind most widely used IDS, classification of various environments of evaluation using NSL-KDD, CICIDS2017, and UNSW-NB15 benchmarks, and development of current indexes of efficiency are performed. In addition, the most important

DOI: Pending

Published by
World Academic Press

problems such as the problem of imbalance of data, the problem of attacks against static models, and the problem of interpretability of models have been discussed.

Keywords: Cybersecurity, machine learning, intrusions detection, classification of threats, anomaly detection, cyberspace.

I. INTRODUCTION

In the era of high dependence on digitally interconnected systems, cybersecurity has become a fundamental pillar of the computational landscape. Organizations, governmental institutions, and educational systems utilize extensive networked environments for communication, data storage, and financial transactions. In this digital transformation, threats such as phishing, malware injection, Denial-of-Service (DoS), and ransomware are increasingly sophisticated, posing significant risks to network integrity.

As these threats are increasing in both magnitude and sophistication, more intelligent and adaptable security mechanisms are now needed more than ever.

Traditional cybersecurity systems, by and large, rely on the concept of predetermined rules or signature-matching approaches to identify well-known threats. While such approaches can detect previously identified attacks, they generally falter in cases of zero-day attacks, new malware variants, or fast-changing invasion patterns. This gap has given rise to various data-driven techniques for the identification of intrusion activity even without having clear attack signature. Machine learning offers one such technology by allowing systems to learn from historical data, uncover latent patterns, and predict malicious actions.

Machine Learning has been deployed for various cyber-security issues like network intrusion detection, malware classification, spam filtering, phishing attacks, fraud detection, user activity modeling, etc.

Out of these areas, the intrusion detection system is one of the main areas of focus, as it identifies unauthorized or harmful activity on the computer networks. In a machine learning-

based intrusion detection system, relevant features of network traffic are extracted and a model is created to identify network events as normal or malicious activity, resulting in the improved accuracy, reducing manual effort and also faster response time to potential emerging threats as compared to other methods. However, machine learning models themselves are not devoid of potential challenges when deployed in security applications.

Issues such as the presence of imbalanced and noisy data, high-dimensionality and the possibility of adversarial attack where data is manipulated to mislead classifier may arise, negatively impacting the accuracy of such models. Another issue, especially, is the interpretation of the complex machine learning model's decision which may lead to difficulties in justifying their actions. Such issues therefore necessitate rigorous study on both potential strengths and weaknesses of the machine learning based security system.

This research is aimed at addressing this issue by exploring the use of machine learning for intrusion detection and classification of cybersecurity threats.

Our primary goal is to assess the role of machine learning based approaches in improving the threat detection effectiveness in computer environment and the potential practical implications of these methods in actual deployment. The paper is structured as follows: Related work is discussed in section II. The methodology used in this work is discussed in section III. The experimental setup is discussed in section IV.

Section V describes the results and discussion.

Finally, section VI concludes this paper.

II. RELATED WORKS

The intersection of machine learning and intrusion detection has generated significant academic output over the past decade. Research has progressively moved away from rigid signature engines toward adaptive algorithms capable of handling high-velocity telemetry data. Analysts have demonstrated that while ML automates threat diagnostics, baseline performance remains bounded by data quality and feature-engineering choices.

III. METHODOLOGICAL FRAMEWORK OF ML-BASED IDS

Rather than presenting an unverified experimental implementation, this section defines the unified, standard methodological pipeline established within modern literature to construct an

operational machine learning-based IDS. The classic workflow is structured as a sequential series of discrete operations, mapped from raw ingestion to model verification.

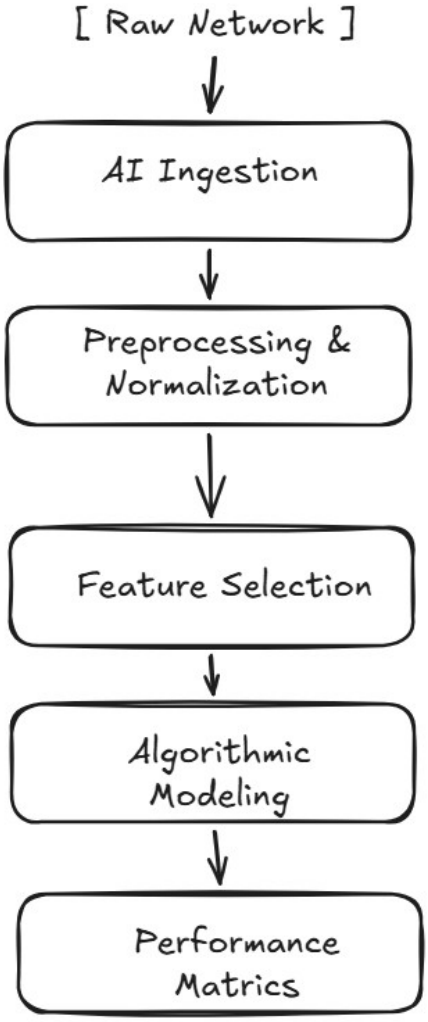

Figure 1: Standard machine learning-based intrusion detection framework pipeline.

1. Dataset Collection and Ingestion

The baseline phase requires ingesting standardized, high-fidelity network datasets generated by specialized research laboratories (e.g., NSL-KDD, UNSW-NB15, CICIDS2017). These repositories contain pre-labeled observations comprising thousands of benign transactions interspersed with distinct categories of cyber exploits,

serving as the ground-truth foundation for supervised and semi-supervised training regimes.

2. Preprocessing and Normalization

Raw network logs cannot be directly interpreted by machine learning cost functions. The preprocessing pipeline standardizes data via four primary steps:

- Data Cleaning: Eradicating or imputing missing, corrupt, or null values within telemetry fields.
- Categorical Feature Encoding: Converting non-numeric network protocols (e.g., TCP, UDP, ICMP, HTTP) into discrete numerical vectors using strategies such as One-Hot Encoding.
- Feature Scaling: Normalizing numerical variables (such as packet length or duration) to a standardized range using Min-Max normalization or Z-score standardization. This prevents attributes with massive scales from disproportionately dominating optimization algorithms.
- Imbalance Correction: Utilizing synthetic sampling methods during the preprocessing phase to stabilize heavily skewed class distributions.

Modern intrusion detection datasets often contain redundant or non-informative features, which can impede computational efficiency and classification accuracy. To optimize feature sets, researchers typically employ three primary methodologies:

- Filter Methods: We use math to look at each feature on its own and get rid of the ones that are not very useful before we start training our model. For example we can use Information Gain or Chi-Square tests to score each feature.
- Wrapper Methods: We use a classifier to try out different groups of features and see which ones work best. We can. Remove features based on how well they help us make predictions. For example we can use Recursive Feature Elimination.
- Embedded Methods: Some algorithms can pick the features naturally when they are trying to find the best solution. For example LASSO regularization and Random Forest feature importance scoring can help us find the important features.

IV. MATHEMATICAL AND ALGORITHMIC FOUNDATIONS

To understand how an IDS determines anomalies, we must analyze the core mathematical and logical reasoning that dictates how standard machine learning classifiers construct decision boundaries:

A. Decision Trees (DT)

Decision Trees recursively partition the multi-dimensional feature space based on attribute thresholds. The selection of the splitting attribute is governed by optimization criteria such as Shannon Entropy, H(X), or Gini Impurity, G(p):

$$H(X) = -\Sigma_{(i=1 \text{ to } c)} p_i \log_2(p_i)$$

$$G(p) = 1 - \Sigma_{(i=1 \text{ to } c)} p_i^2$$

Where **p_i** is the probability of a network record belonging to class **i** within a total of **c** distinct threat classes. The algorithm selects the feature threshold that maximizes information gain, minimizing impurity in the resulting child nodes to separate benign packets from malicious anomalies.

B. Random Forests (RF)

An extension of the Decision Tree, Random Forest is an ensemble architecture that constructs a collection of independent decision trees during training. It injects variance through bootstrap aggregation ("bagging") and random feature selection. The final classification prediction $\hat{y}$ for an incoming network packet is derived via majority voting across **B** individual trees:

Mathematically, this ensemble approach reduces the variance of the overall model without increasing its bias, preventing the overfitting issues common to single decision trees when handling complex network environments.

C. Support Vector Machines (SVM)

Support Vector Machines project input network feature vectors **x** into higher-dimensional spaces to discover an optimal separating hyperplane that maximizes the geometric margin between normal traffic and cyberattacks. The decision boundary is defined by the hyperplane equation:

$$w^T x + b = 0$$

To handle non-linear structural distributions, the optimization problem is solved using kernel functions **φ(x)**. The optimization objective minimizes regularization weights while limiting classification errors through slack variables **ξ_i**:

$$\text{minimize } (1/2)\|w\|^2 + C \Sigma \xi_i$$

$$\text{subject to } y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

This allows the system to establish clean, robust boundaries separating normal operations from complex, malicious traffic anomalies.

D. K-Nearest Neighbors (k-NN)

The k-NN classifier is a non-parametric, instance-based learning algorithm that bypasses explicit training phases. It classifies an unknown network packet **x** by calculating its geometric proximity to historical reference samples stored in memory. The spatial distance is conventionally computed via the Euclidean Distance metric:

$$d(x, y) = \sqrt{\sum_{k=1}^{d}(x_k - y_k)^2}$$

The input sample is assigned to the majority threat class present among its k closest neighbors within the multi-dimensional feature space.

V. ANALYSIS OF EXPERIMENTAL FRAMEWORKS AND BENCHMARKS

1. Benchmark Reference Datasets

Academic literature relies on a core group of publicly available datasets to test and cross-validate the efficiency of ML classification structures:

Dataset	Generation Year	Predominant Protocol Layer	Key Threat Modalities Represented
NSL-KDD	2009	Network / Transport (IP, TCP)	DoS, U2R (User to Root), R2L (Remote to Local), Probing

UNSW-NB15	2015	Network / Transport / Application	Fuzzers, Analysis, Backdoors, Exploits, Generic, Reconnaissance
CICIDS2017	2017	PCAP / Multi-Layer Netflow	Brute Force, Heartbleed, Botnets, DoS, DDoS, Web Attacks

2. VALIDATION AND PARTITIONING STRATEGIES

To prevent overfitting and ensure models generalize well to live network environments, standard validation protocols enforce strict division configurations:

- Train/Test Splitting: Segmenting data into distinct training sets (e.g., 70-80%) to compute internal model parameters, leaving the remaining portion completely unseen for validation testing.
- K-Fold Cross-Validation: Dividing datasets into **K** equal subsets. The system trains iteratively on **K-1** blocks while validating on the remaining block, repeating the process **K** times to stabilize performance matrics against localized bias.

3. COMPREHENSIVE EVALUATION FRAMEWORKS

Relying solely on standard classification accuracy can produce highly misleading results due to the massive numerical dominance of benign traffic in real networks. Comprehensive research frameworks measure performance through a collection of matrics derived from the Confusion Matrix:

- Accuracy: The overall proportion of correctly classified instances (both benign and malicious).
- Precision (Positive Predictive Value): The ratio of true malicious alerts relative to all generated alerts, showing how well the system avoids false positives.

$$\text{Precision} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Positives(FP)}}$$

- Recall (Sensitivity / Detection Rate): The percentage of actual attacks successfully caught by the system, indicating how well it avoids missing active threats.

- F1-Score: The harmonic mean of precision and recall, serving as a balanced performance metric for highly skewed datasets.

- False Alarm Rate (FAR / False Positive Rate): The percentage of safe, benign transactions flagged as malicious threats, which creates unnecessary work for security analysts.

VI. COMPARATIVE PERFORMANCE ANALYSIS AND DISCUSSION

Recent empirical literature demonstrates that machine learning classifiers achieve robust detection capabilities within controlled environments. Performance metrics consistently indicate high classification accuracy across standardized datasets such as UNSW-NB15 and CICIDS2017. Ensemble architectures, particularly Random Forest and Gradient Boosting machines, frequently demonstrate detection accuracy exceeding 99% for standard threat profiles. This represents a significant performance improvement over traditional signature-based detection, which fails to generalize to novel attack patterns.

However, laboratory performance benchmarks often overestimate operational efficacy. Analysis of the literature reveals critical tradeoffs when deploying these models in real-world scenarios:

1. The Imbalanced Detection Paradox: While models excel at identifying high-volume threats (e.g., standard DoS/DDoS), they exhibit significantly reduced sensitivity toward rare, low-frequency variants such as User-to-Root (U2R) or localized backdoor exploits. These threats are frequently obscured within large volumes of benign traffic, necessitating a prioritization of precision, recall, and F1-score over raw accuracy metrics to ensure operational viability.

2. Computational Overhead vs. Deployment Scale: Complex architectures, such as CNN-LSTM chains, provide superior identification of non-linear attack signatures but impose substantial computational costs that limit their feasibility in edge-computing environments, such as IoT devices or industrial automation nodes. Conversely, lighter models—including

optimized Decision Trees and linear Support Vector Machines—deliver competitive performance with a lower resource footprint, offering greater ease of maintenance and faster update cycles in distributed deployments.

The following table summarizes the comparative performance characteristics of common machine learning architectures identified in the literature:

Model Architecture	Strengths	Limitations	Optimal Use-Case
Random Forest	High accuracy; non-linear patterns	Computationally intensive	General-purpose IDS
SVM	High-dimensional efficiency	Noise sensitivity	Binary classification

VII. CONCLUSION

The integration of machine learning into operational cybersecurity infrastructures necessitates robust mitigation strategies against algorithmic vulnerabilities. To ensure the reliability of these defensive systems, specific structural challenges must be addressed through advanced architectural refinements.

1. Mitigation of Data Imbalance
In network environments, malicious traffic constitutes a minority class, often representing less than 1% of total data. Standard optimization algorithms, focused on global error reduction, inherently bias towards majority class classification, thereby degrading detection efficacy for minority attack vectors.

To improve detection sensitivity, two primary methodologies are utilized: first, cost-sensitive learning, which applies higher penalty weights to false negative classifications of malicious traffic; and second, synthetic data augmentation via SMOTE or ADASYN, which restores class equilibrium by generating synthetic minority samples within the feature space prior to model training.

2. Defense Against Adversarial Exploitation
Machine learning models are susceptible to adversarial evasion attacks. Attackers exploit this by applying targeted perturbations—such as latency injections—to network traffic, thereby misclassifying malicious payloads as benign.

Figure 2: Conceptual representation of adversarial evasion in intrusion detection classifiers.

Mitigation frameworks utilize adversarial training, where synthetic attack vectors are integrated into the training corpus, and defensive distillation techniques, which obscure predictive probabilities to hinder evasion attempts.

This study concludes that while machine learning significantly outperforms traditional signature-based methods in classifying malicious network traffic—as demonstrated across the NSL-KDD, UNSW-NB15, and CICIDS2017 benchmarks—its operational success depends on the deployment of reliable feature selection, ensemble architectures, and robust adversarial protection mechanisms. Future research will focus on implementing adaptive systems capable of real-time threat identification within resource-constrained edge environments.

VIII. REFERENCES

1. Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F. (2022). Cybersecurity threats and their mitigation approaches using machine learning—A review. *Journal of Cybersecurity and Privacy*, *2*(3), 527–555. https://doi.org/10.3390/jcp2030027
2. Dasgupta, D., Akhtar, Z., & Sen, S. (2022). Machine learning in cybersecurity: A comprehensive survey. *The Journal of Defense Modeling and Simulation*, *19*(1), 57–106. https://doi.org/10.1177/1548512920951275
3. Du, J. (2023). NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. *IEEE Access*, *11*, 24681–24695. https://doi.org/10.1109/ACCESS.2023.3254181
4. Gaddah, F. W. (2024). Cyber threat detection using machine learning. *In Proceedings of the 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 412–417. https://doi.org/10.1109/MI-STA61439.2024.10553316
5. Gopalsamy, M., Patil, S., & Dudhankar, V. (2025). Advancements in cybersecurity: Evaluating machine learning approaches for detecting cyber attacks. *In Proceedings of the 2025 3rd International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 114–119.
6. Hegde, R., Likitha, S., & Natesh, M. (2024). Intrusion detection system using machine learning based on NSL-KDD dataset. *In Proceedings of the International Conference on Recent Advances in Computer Science and Engineering (ICRACE)*, 88–93.
7. V. Aakash, D. (2025). Comparative evaluation of machine learning models for intrusion detection in communication networks. *In Proceedings of SPIE 13660, International Conference on Information Technology*, 136600E. https://doi.org/10.1117/12.3041122
8. Žada, F., & Antonić, M. (2024). Network traffic intrusion detection. *In Proceedings of the 32nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 301–305. https://doi.org/10.23919/SoftCOM62812.2024.10722354
9. Zhang, J., & Pan, L. (2021). Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica*, *8*(3), 563–575. https://doi.org/10.1109/JAS.2021.1003874