EuroVantage
Journal of Artificial Intelligence

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

Machine Learning-Based Intrusion Detection Systems for Enterprise Security Architectures

[1] Jabulani Khumalo, [2] Meera Kapoor

[1] University of South Africa, Pretoria, South Africa

[2] IIT Madras, Chennai, India

Corresponding Author: jabulani.khumalo@interviauniversity.com

Abstract

The rapid expansion of enterprise networks, coupled with the increasing sophistication of cyber threats, has rendered traditional signature-based intrusion detection systems (IDS) largely inadequate for modern security architectures. This paper explores the integration of machine learning (ML) techniques into intrusion detection frameworks specifically designed for large-scale enterprise environments. We examine supervised, unsupervised, and semi-supervised learning models—including random forests, support vector machines, autoencoders, and deep neural networks—and evaluate their efficacy in detecting zero-day attacks, polymorphic malware, and insider threats. A comparative analysis of ML-based IDS against conventional systems is presented, focusing on key performance metrics such as detection rate, false positive rate, computational overhead, and adaptability to evolving attack vectors. Furthermore, the paper addresses architectural challenges unique to enterprises, including data imbalance, real-time processing constraints, and integration with existing security information and event management (SIEM) systems. Findings indicate that hybrid ML models, particularly those combining anomaly detection with ensemble learning, significantly enhance detection accuracy while maintaining acceptable latency for high-throughput enterprise networks. We conclude with a set of best practices for deploying ML-based IDS within defense-in-depth strategies and propose future research directions in adversarial machine learning and federated learning for distributed enterprise architectures.

Keywords: Intrusion Detection System (IDS), Machine Learning, Enterprise Security Architecture, Anomaly Detection, Network Traffic Analysis, Deep Learning, Cybersecurity

I. Introduction

The contemporary enterprise security landscape faces an unprecedented challenge: cyberattacks are no longer conducted solely by lone hackers but by organized threat groups employing automated tools, zero-day exploits, and advanced persistent threats (APTs). Traditional intrusion detection systems, which rely on signature-based or rule-based mechanisms, operate on the principle of matching known attack patterns against a database of signatures. While effective against previously

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

documented threats, these systems fail catastrophically when encountering novel attacks or subtle variations of existing malware. A single zero-day exploit can bypass signature-based IDS entirely, leaving enterprise networks vulnerable for extended periods. Moreover, the explosion in network traffic volume—driven by cloud migration, IoT devices, and remote work—has overwhelmed rule-based systems, causing alert fatigue and missed detections. Consequently, enterprise security architects have turned to machine learning (ML) as a paradigm shift that enables IDS to learn normal and malicious behavior patterns directly from data, thereby generalizing to unseen threats. This paper argues that ML-based IDS are not merely incremental improvements but foundational components of next-generation enterprise security architectures, provided they are designed with careful attention to scalability, interpretability, and adversarial resilience.

The integration of ML into enterprise IDS introduces both promise and complexity. Unlike academic or experimental settings, enterprise environments demand real-time or near-real-time analysis of network flows, system logs, and endpoint telemetry. False positives, which might be tolerable in a research lab, can paralyze security operations centers (SOCs) by triggering hundreds of irrelevant alerts per minute, leading analysts to ignore genuine threats. Conversely, false negatives—undetected intrusions—can result in data breaches, financial losses, and regulatory penalties. Therefore, selecting appropriate ML algorithms and feature engineering techniques is not a purely theoretical exercise but a business-critical decision. This paper systematically reviews state-of-the-art ML models for intrusion detection, evaluates their suitability for enterprise deployment across metrics like precision, recall, F1-score, and inference latency, and proposes a reference architecture that integrates ML-based detection layers with existing enterprise security controls. By synthesizing findings from recent literature and real-world deployment case studies, we aim to provide a actionable roadmap for security engineers and architects seeking to modernize their intrusion detection capabilities.

II. Limitations of Conventional IDS in Enterprise Settings

Signature-based IDS, such as Snort and Suricata, operate by matching network packets or log entries against a database of predefined attack signatures. This approach excels at detecting known exploits, but it suffers from three fundamental limitations when applied to enterprise architectures. First, signature databases must be continuously updated, creating a window of vulnerability between the discovery of a new attack and the release of a signature. In modern high-speed attack campaigns, this window can be exploited within minutes, as seen in ransomware outbreaks like WannaCry and NotPetya. Second, signature-based systems cannot identify attacks that use encryption, polymorphism, or minor modifications to evade pattern matching[1]. For example, a metamorphic virus that changes its binary signature each time it replicates will never match a static signature, yet it can propagate unchecked across an enterprise. Third, the sheer volume of unique applications, protocols, and custom enterprise services generates an astronomical number of possible normal behaviors; maintaining signatures for every legitimate process is infeasible. These limitations have led to what security researchers call the ―detection gap‖—the growing divergence between what enterprises need to detect and what signature-based tools can actually identify.

Anomaly-based IDS were introduced to address these gaps by establishing a baseline of ―normal‖ network or system behavior and flagging deviations as potential intrusions. Early anomaly detection systems used statistical methods like mean and standard deviation of packet sizes or connection rates.

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

However, these simplistic models generated unacceptably high false positive rates in dynamic enterprise environments where normal behavior changes due to software updates, user activity patterns, or seasonal traffic bursts. A marketing department launching a legitimate ad campaign, for instance, could trigger thousands of anomaly alerts due to sudden spikes in outbound connections. Furthermore, anomaly-based systems often lack the ability to explain why an event was flagged, forcing SOC analysts to manually investigate each alert. This combination of high noise and low interpretability led many enterprises to disable or severely tune down anomaly detection features, effectively reverting to signature-only protection. Machine learning addresses these weaknesses by learning complex, non-linear relationships in high-dimensional data, distinguishing between benign anomalies (e.g., a scheduled backup) and malicious anomalies (e.g., data exfiltration) with far greater accuracy.

III. Machine Learning Models for Intrusion Detection

Supervised learning remains the most widely researched category for ML-based IDS, primarily due to the availability of labeled datasets such as KDD Cup 99, NSL-KDD, CIC-IDS-2017, and UNSW-NB15. In this paradigm, a model is trained on historical network traffic or host logs where each record is labeled as ―normal‖ or a specific attack class (e.g., DoS, probe, R2L, U2R). Algorithms such as random forests, gradient boosting machines (XGBoost, LightGBM), and support vector machines (SVMs) have demonstrated strong performance, often achieving detection rates above 95% on benchmark datasets[2]. Random forests are particularly attractive for enterprise use because they provide built-in feature importance rankings, helping security analysts understand which attributes (e.g., packet length, protocol type, connection duration) contribute most to a detection decision. However, supervised learning suffers from a critical weakness: it requires large, accurately labeled datasets that represent all possible attack types. In real-world enterprises, labeling network traffic at scale is prohibitively expensive, and emerging attacks have no labeled examples, causing supervised models to fail silently[3].

Unsupervised and semi-supervised learning techniques circumvent the labeling bottleneck. Autoencoders, a type of neural network trained to reconstruct input data, learn a compressed representation of normal network behavior. During inference, a high reconstruction error indicates an anomaly, potentially signaling an intrusion. This approach has proven effective for detecting zero-day exploits and subtle insider threats—such as an employee exfiltrating small amounts of data over weeks—without requiring attack labels. Clustering algorithms like DBSCAN and self-organizing maps can also group similar network flows and flag outliers as suspicious. The primary challenge with unsupervised methods is controlling false positives: because no ground truth is provided during training, the model may classify any deviation—including legitimate changes in user behavior—as malicious. Semi-supervised approaches, which train on only normal data and then detect deviations, offer a pragmatic middle ground. Deep learning variants, including convolutional neural networks (CNNs) applied to flow visualizations and long short-term memory (LSTM) networks for time-series analysis of sequential traffic, have further improved detection of slow-and-low attacks that spread over days or weeks. Nevertheless, deep models require substantial computational resources and careful hyperparameter tuning, making them less suitable for resource-constrained edge devices within distributed enterprise architectures.

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

IV. Architectural Integration with Enterprise Security Stacks

Deploying an ML-based IDS within an enterprise security architecture is not merely a matter of running a Python script on a network tap[4]. A robust integration requires careful placement of detection sensors, data preprocessing pipelines, model serving infrastructure, and orchestration with existing security tools. Most enterprises adopt a hybrid detection architecture comprising network-based IDS (NIDS) sensors at critical chokepoints (e.g., internet gateways, data center switches, cloud VPCs) and host-based IDS (HIDS) agents on servers and endpoints. ML models can be deployed in two modes: inline (where traffic is blocked or allowed in real-time based on model decisions) or out-of-band (where alerts are sent to a SIEM for correlation and manual investigation). For latency-sensitive applications, inline ML inference requires optimized models (e.g., pruned decision trees or quantized neural networks) and hardware acceleration (GPUs or FPGAs) to keep packet processing under tens of microseconds per flow[5].

The integration layer must also handle feature extraction in real-time. Raw network flows from tools like Zeek (formerly Bro) or NetFlow provide hundreds of potential features—source/destination IPs, ports, protocol, packet counts, byte volumes, flags, and inter-arrival times[6]. Many of these features are categorical (e.g., protocol) or high-cardinality (e.g., IP addresses), requiring encoding schemes like one-hot or feature hashing. A common enterprise pattern is to implement a two-stage pipeline: a lightweight statistical preprocessor that computes aggregated flow features over time windows (e.g., 1-second, 10-second, 60-second bins), followed by an ML inference engine that scores each window. This approach reduces the computational burden and smooths out transient spikes. The output of ML-based IDS should feed directly into the enterprise's security orchestration, automation, and response (SOAR) platform, enabling automated playbooks—for example, when a model detects beaconing behavior indicative of command-and-control traffic, the SOAR can automatically isolate the affected host, revoke its network access, and create a ticket for analyst review. Without such integration, ML alerts risk becoming just another silo of unactionable data.

V. Performance Evaluation and Benchmarking

Evaluating ML-based IDS for enterprise deployment requires metrics beyond simple accuracy, given the severe class imbalance in real network traffic (normal events typically outnumber attacks by ratios of 1000:1 or higher). Precision, recall, and the F1-score are more informative: high recall ensures that true attacks are identified, while high precision minimizes analyst wasted time on false alerts. The receiver operating characteristic (ROC) curve and the area under it (AUC) provide a threshold-independent measure of separability. However, enterprises must also measure practical operational metrics: mean time to detect (MTTD), mean time to respond (MTTR), and false positive rate per million events (FPR/1M). A model that achieves 99% recall but generates 0.1% false positives on 10 million daily events would produce 10,000 false alerts per day—overwhelming a typical SOC. Therefore, many enterprises tune models for extremely low false positive rates (e.g., 0.001%) even at the cost of some reduction in recall, then layer multiple models or use human-in-the-loop validation.

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

Recent benchmark studies on modern datasets (CSE-CIC-IDS-2018, TON-IoT) reveal that no single ML model dominates across all attack categories. Random forests perform exceptionally well on discrete protocol-based attacks (e.g., port scans, web application exploits) but struggle with subtle behavioral anomalies like credential stuffing over encrypted channels. Deep autoencoders excel at detecting rare insider threats but require careful threshold selection to avoid over-sensitivity. Ensemble methods that combine multiple models—for example, a gradient-boosted tree for signature-like attacks and an isolation forest for outliers—achieve the highest overall F1-scores (0.96–0.98) but introduce higher inference latency (50–200 ms per flow). For real-time enterprise needs, a tiered architecture is recommended: a lightweight rule-based filter drops obviously benign traffic (e.g., internal DNS), an ML classifier analyzes remaining flows with a latency budget of 10 ms, and a deeper forensic model inspects only flagged events offline. Computational cost must also be considered: training deep neural networks on 10 million flow records can require GPU hours, while inference of a quantized random forest may run on a single CPU core. Enterprises must balance detection capability against hardware and cloud costs, often opting for hybrid deployments where cloud-based models handle peak loads and edge-based models provide baseline protection.

VI. Conclusion

Machine learning has fundamentally reshaped the intrusion detection landscape, offering a path beyond the limitations of signature-based and statistical anomaly systems that have dominated enterprise security for decades. This paper has demonstrated that ML-based IDS, when properly architected, can detect zero-day exploits, polymorphic attacks, and subtle insider threats that would otherwise remain invisible. However, successful deployment requires far more than selecting an algorithm with high benchmark accuracy. Enterprises must design end-to-end pipelines that handle real-time feature extraction, model inference within strict latency budgets, seamless integration with SIEM and SOAR platforms, and continuous adaptation to evolving network behavior. The challenges of adversarial evasion, concept drift, interpretability, and data imbalance remain active research frontiers, but practical solutions—such as hybrid ensembles, tiered architectures, and continuous feedback loops—are already enabling early adopters to strengthen their security postures. Ultimately, ML-based IDS should be viewed as a critical layer within a defense-in-depth strategy, complementing firewalls, endpoint protection, and zero-trust network access controls. As adversarial machine learning techniques mature and federated learning enables privacy-preserving collaboration across enterprise boundaries, we anticipate that ML-based IDS will evolve from a cutting-edge option to a mandatory component of all serious enterprise security architectures. The question is no longer whether to adopt machine learning for intrusion detection, but how to do so responsibly, cost-effectively, and resiliently.

References:

Pages: 90-95
Volume-I, Issue-III, 2025
https://evjai.com/index.php/evjai

[1] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, 2012, vol. 1: IEEE, pp. 647-651.

[2] W. R. Claycomb and A. Nicoll, "Insider Threats to Cloud Computing: Directions for New Research Challenges," in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*, 2012: IEEE, pp. 387-394.

[3] P. T. Endo, G. E. Gonçalves, J. Kelner, and D. Sadok, "A survey on open-source cloud computing solutions," in *Brazilian Symposium on Computer Networks and Distributed Systems*, 2010.

[4] I. Ivanov, C. Maple, T. Watson, and S. Lee, "Cyber security standards and issues in V2X communications for Internet of Vehicles," 2018.

[5] K. Hwang, S. Kulkareni, and Y. Hu, "Cloud security with virtualized defense and reputation-based trust mangement," in *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*, 2009: IEEE, pp. 717-722.

[6] B. R. Kandukuri, V. R. Paturi, and A. Rakshit, "Cloud security issues," in *Services Computing, 2009. SCC'09. IEEE International Conference on*, 2009: IEEE, pp. 517-520.