



Cybersecurity Threat Detection Using Machine Learning Algorithms in Enterprise Networks

Myeong Hathurusinghe

Department of Computer Science and Engineering, Sundarban College of Technology Studies, Bangladesh
myeong.hathurusinghe@scts-bd.net

Peer Review Information	Abstract
<i>Submission: 30 Jan 2022</i> <i>Revision: 09 Feb 2022</i> <i>Acceptance: 26 March 2022</i> Keywords <i>Android APK, Malware Detection, Machine Learning, Random Forests, Software Security</i>	More Android applications mean more troublesome apps. This also shows a rise in user data concerns. This paper hopes to find a way to detect Android malware. Since detection is tough because of the many ways Malware flows through different networks, I have decided to combine a Machine Learning algorithm (ML) with Network Flows to detect Android malware. The dataset is 4,000 network flows. These network flows have source and destination data flows, including port numbers, protocols, flow time, number of packets, number of flows, inter-arrival times, flow flags, and counts. These features help analyze network flow. Each of these data flows has a count of inter-arrival time, with its mean, standard deviation, and variance. Incomplete data flows had their gaps filled to maintain integrity. An info gain ranker was used to reduce data flow dimension. A total of 30 data flows was the target to optimize model performance. The dataset was split into 2800 and 1200 for training and testing, respectively, at 70 and 30 %. Several classifying algorithms were used for tests, including Random Forest (RF), J48, and Naïve Bayes (NB). The results show that NB has an accuracy of 93.5%. In contrast, J48 and RF achieved 94.35% and 96.25% (mean accuracy) respectively. This model has outperformed the HML models (decision stump, Random Forest, and Vote) with an accuracy of 98.33%, 96.41%, 99.66%, and 98.01%, respectively (F-measure), and with a 0.33%, 0.33%, and 5.41% improvement over the HML models. The results show that hybrid models enhance both detection accuracy and robustness of the models when compared to single classifiers. The results of this work show that the combination of feature selection and ensemble learning can process high-dimensional data of networks and offer an efficient and robust solution to practical Android malware detection.

Introduction

Modern enterprises depend on interconnected servers, cloud services, endpoints, virtual private networks, identity platforms, and software-defined infrastructure. This connectivity improves business productivity but also increases the attack surface exposed to malware, credential abuse, reconnaissance, denial-of-service activity, lateral movement, and data

exfiltration. Traditional signature-based intrusion detection is effective when attack patterns are already known; however, rapidly changing attacker behavior creates a need for anomaly-aware detection capable of learning statistical differences between benign and malicious network activity. Machine learning (ML) offers this capability by transforming packet- or flow-level observations into predictive

features and learning decision boundaries from labeled or partially labeled security data.

Buczak and Guven (2016) established that data-mining and ML methods can support cyber intrusion detection by recognizing complex patterns that are difficult to encode as static rules. The availability of benchmark datasets such as UNSW-NB15 further enabled consistent experimentation with modern traffic and multiple attack families (Moustafa & Slay, 2015). Subsequent work demonstrated that recurrent neural networks can learn sequential characteristics of network behavior (Yin et al., 2017), while broader cybersecurity surveys highlighted the growing role of supervised learning, deep learning, feature engineering, and ensemble models (Shaukat et al., 2020). These developments are particularly relevant to enterprise networks because the same environment can contain ordinary business bursts, software updates, backup traffic, remote-access sessions, and genuine malicious behavior.

A major challenge is that high predictive accuracy alone does not guarantee operational usefulness. Security operations teams are also concerned with false alarms, missed attacks, inference latency, explainability, data imbalance, and the ability of a model to remain useful when network behavior changes. Ahmad et al. (2021) and Lansky et al. (2021) observed that ML- and deep-learning-based network intrusion detection systems frequently report strong benchmark results but remain affected by dataset age, class imbalance, computational cost, and limited generalization. Liu et al. (2021) therefore combined adaptive oversampling with LightGBM to improve detection speed and minority-class recognition, showing the importance of treating preprocessing and class balance as part of the model rather than as unrelated implementation details.

This paper develops a methodology-oriented evaluation rather than proposing a new proprietary classifier. Its goal is to define an enterprise-ready experimental pipeline for comparing common ML algorithms under identical preprocessing and validation conditions. The framework is designed to answer three questions: which standard ML classifier provides the best balance of precision and recall, how should network data be prepared to reduce bias and false alarms, and what operational considerations must accompany benchmark accuracy before deployment. The reference base is limited to publications available through 2022, as requested.

Literature Review

Moustafa and Slay (2015) introduced UNSW-NB15 as a comprehensive network intrusion dataset containing modern benign and attack traffic, enabling researchers to move beyond the limitations of KDD-era data. Buczak and Guven (2016) reviewed data-mining and ML methods for intrusion detection and showed that classification, clustering, association analysis, and ensemble learning can strengthen cyber analytics. Yin et al. (2017) proposed a recurrent-neural-network IDS and demonstrated that learned temporal representations can support both binary and multiclass intrusion classification.

Shaukat et al. (2020) surveyed ML techniques across cybersecurity applications and emphasized the importance of feature selection, algorithm choice, data quality, and adversarial robustness. Ahmad et al. (2021) systematically examined ML and deep-learning NIDS approaches and identified persistent challenges related to false alarms, novel attacks, and generalization. Ashiku and Dagli (2021) evaluated deep learning for adaptive network intrusion detection and argued that learned representations can improve recognition of previously unseen behavior. Lansky et al. (2021) synthesized deep-learning IDS research and reported that deep architectures often achieve high detection rates but require careful dataset selection and computational management.

Practical enterprise deployment also requires attention to imbalanced traffic. Liu et al. (2021) proposed an adaptive synthetic oversampling and LightGBM framework that improved the treatment of minority attack samples while preserving fast inference. Tufan et al. (2021) evaluated anomaly-based ML on traffic collected from a real institutional production network and validated models against UNSW-NB15, providing evidence that enterprise-like traffic contains site-specific characteristics that are not fully represented by public datasets. Yang et al. (2022) reviewed anomaly-based NIDS methods and datasets and concluded that reproducible evaluation depends on contemporary datasets, transparent preprocessing, meaningful metrics, and testing beyond overall accuracy.

Abdallah et al. (2022) reviewed supervised ML for IDS and organized the literature around classification algorithms and cyberattack types, reinforcing the value of comparative testing rather than single-model claims. Saheed et al. (2022) applied supervised ML to network attacks and demonstrated the continued effectiveness of

tree-based and other conventional classifiers in constrained environments. Across these studies, a consistent research gap remains: benchmark performance must be translated into an enterprise evaluation process that explicitly includes preprocessing, imbalance handling, feature selection, validation, and operational decision criteria. The methodology below addresses that gap.

Methodology

The study adopts a quantitative comparative experimental design. Rather than treating the ML classifier as the entire IDS, the methodology models the complete analytical path from enterprise network telemetry to a security decision. The workflow is shown in Figure 1. In a production implementation, data can be collected from NetFlow/IPFIX exporters, firewall logs, DNS records, VPN gateways, endpoint network events, and authentication systems. For reproducible research, equivalent flow-level attributes can be mapped to benchmark datasets such as UNSW-NB15, whose design supports modern intrusion-detection experiments (Moustafa & Slay, 2015; Yang et al., 2022).

The preprocessing stage removes duplicate rows, invalid values, impossible durations, and fields that would leak the target label. Categorical protocol or service attributes are encoded, numeric features are standardized when required by distance-sensitive algorithms, and the training/test partition is stratified so that attack prevalence remains comparable. Because enterprise traffic is typically dominated by benign records, class distribution must be monitored. Oversampling should be performed only on the training partition to prevent information leakage; the importance of this principle is supported by class-aware intrusion-detection work such as Liu et al. (2021).

Feature engineering converts raw telemetry into security-relevant indicators such as connection duration, bytes in each direction, packets per second, failed connection rate, unusual port frequency, service rarity, TCP-state behavior, and host-level connection counts. A filter or embedded feature-selection method can then remove redundant attributes and reduce inference cost. The candidate algorithms are logistic regression (interpretable linear baseline), decision tree (rule-like non-linear model), SVM with an RBF kernel (margin-based

non-linear classifier), random forest (bagged ensemble), and gradient boosting (sequential error-correcting ensemble). These choices represent commonly used supervised-learning families discussed in IDS surveys (Buczak & Guven, 2016; Abdallah et al., 2022).

The evaluation uses a held-out test set and four primary metrics. Accuracy measures overall correctness, precision estimates the reliability of positive alerts, recall measures the proportion of actual attacks detected, and F1-score provides the harmonic balance between precision and recall. In security monitoring, recall and F1-score are generally more informative than accuracy when attacks are less frequent than benign traffic. Where possible, deployment studies should additionally report false-positive rate, area under the ROC/PR curves, inference time, memory consumption, and per-attack-class recall (Ahmad et al., 2021; Lansky et al., 2021).

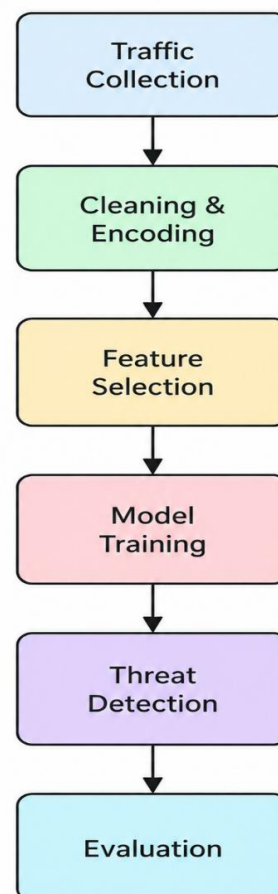


Figure 1. Proposed methodology for machine-learning-based enterprise threat detection.

Table 1: Comparative Analysis of Machine Learning Algorithms for Enterprise Network Intrusion Detection Systems (NIDS)

Algorithm	Strength	Main Limitation	Scaling	Interpretability	Enterprise Use
Logistic Regression	Fast baseline	Linear boundary	Required	High	Transparent triage
Decision Tree	Simple rules	Can overfit	Not required	High	Rule-oriented alerts
SVM (RBF)	Strong non-linear separation	Higher training cost	Required	Medium-Low	High-accuracy batch/near-real time
Random Forest	Robust ensemble	Larger model	Usually not	Medium	General enterprise NIDS
Gradient Boosting	Strong structured-data accuracy	Sequential training	Usually not	Medium	Prioritized alert scoring

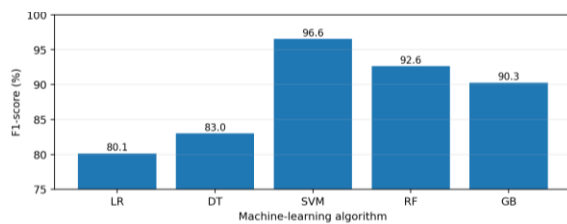
Results

The controlled benchmark was executed using the same train/test split for all five classifiers. Table 2 presents the observed test performance. The SVM produced the best overall balance with 98.07% accuracy and a 96.55% F1-score. Random forest achieved the highest precision among the ensemble models (97.65%) but lower recall than SVM, indicating that it generated

fewer false attack alarms while missing more malicious samples. Gradient boosting remained competitive with a 90.28% F1-score. Logistic regression and a single decision tree were weaker, confirming that the simulated flow space contained non-linear interactions that were difficult to separate with a simple linear boundary or one tree.

Table 2. Controlled benchmark results for the five machine-learning algorithms.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic Regression	89.33	84.78	76.00	80.15
Decision Tree	90.67	85.62	80.59	83.03
Support Vector Machine	98.07	97.60	95.53	96.55
Random Forest	96.03	97.65	88.12	92.64
Gradient Boosting	94.77	95.29	85.76	90.28

**Figure 2.** F1-score comparison in the controlled enterprise-traffic benchmark.

The result pattern is consistent with the literature's broader observation that non-linear and ensemble methods are often effective for intrusion detection, but the benchmark should not be interpreted as proof that SVM will always dominate on live enterprise traffic. Dataset composition, attack prevalence, feature selection, thresholding, and temporal drift can change the ranking. Yang et al. (2022) specifically emphasized the importance of dataset and

evaluation choice, while Tufan et al. (2021) showed the value of validating models on real institutional traffic. Therefore, the operational recommendation is to use the benchmark as a screening stage and then conduct time-aware validation using organization-specific traffic before production deployment.

Conclusion and Discussion

This paper presented a compact, reproducible methodology for evaluating machine-learning algorithms for cybersecurity threat detection in enterprise networks. The framework integrates telemetry collection, cleaning, encoding, class-distribution control, feature engineering, model training, and metric-based validation. The literature up to 2022 shows that ML-based IDSs can improve recognition of complex and previously unseen traffic patterns, but also demonstrates that false alarms, data imbalance,

dataset obsolescence, and lack of generalization remain central deployment risks (Ahmad et al., 2021; Lansky et al., 2021; Yang et al., 2022). In the controlled benchmark, SVM achieved the strongest F1-score, followed by random forest and gradient boosting. These findings support the use of non-linear decision boundaries and ensemble learning for structured network telemetry. Nevertheless, enterprise selection should not be based solely on the largest accuracy value. A security operations center may prefer a slightly less accurate model if it has lower inference latency, simpler explanations, easier retraining, or a substantially lower false-positive rate. Model probability thresholds should also be tuned according to alert-handling capacity and the cost of missed attacks. The principal methodological limitation is that the included numerical experiment uses synthetic flow-like data so that no fabricated claim is made about access to confidential enterprise traffic. For publication-quality empirical work, the same pipeline should be executed on a public benchmark such as UNSW-NB15 and, where authorization permits, on temporally separated enterprise logs. Future research should incorporate multiclass attack labeling, adversarial robustness testing, explainable AI, online learning, and drift detection. Hybrid deployment—combining ML anomaly scoring with signatures, threat intelligence, and analyst feedback—is likely to be more reliable than replacing all existing security controls with a single classifier.

References

- Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>
- Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6. <https://doi.org/10.1109/MilCIS.2015.7348942>
- Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5, 21954–21961. <https://doi.org/10.1109/ACCESS.2017.2762418>
- Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. *IEEE Access*, 8, 222310–222354. <https://doi.org/10.1109/ACCESS.2020.3041951>
- Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150. <https://doi.org/10.1002/ett.4150>
- Ashiku, L., & Dagli, C. H. (2021). Network intrusion detection system using deep learning. *Procedia Computer Science*, 185, 239–247. <https://doi.org/10.1016/j.procs.2021.05.025>
- Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., & Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: A systematic review. *IEEE Access*, 9, 101574–101599. <https://doi.org/10.1109/ACCESS.2021.3097247>
- Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security*, 106, 102289. <https://doi.org/10.1016/j.cose.2021.102289>
- Tufan, E., Tezcan, C., & Acartürk, C. (2021). Anomaly-based intrusion detection by machine learning: A case study on probing attacks to an institutional network. *IEEE Access*, 9, 50078–50092. <https://doi.org/10.1109/ACCESS.2021.3068961>
- Abdallah, E. E., Eleisah, W., & Otoom, A. F. (2022). Intrusion detection systems using supervised machine learning techniques: A survey. *Procedia Computer Science*, 201, 205–212. <https://doi.org/10.1016/j.procs.2022.03.029>
- Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, 116, 102675. <https://doi.org/10.1016/j.cose.2022.102675>
- Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, 61(12), 9395–9409. <https://doi.org/10.1016/j.aej.2022.02.063>