Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 11 Issue 11, November - 2025

Binary Classification Of Malicious Android Traffic Using Static Network Behavior Profiling, CSE-CIC-IDS2018 Dataset And Random Forest Ensemble Learning Approach

Godwill Ajumo Samuel[1]
Department of Computer Engineering Technology
Captain Elechi Amadi Polytechnic, Rumuola
Port Harcourt Rivers State
godwill.samuel@portharcourtpoly.edu.ng

OJEDOKUN Isaac Adewale[2]
Department OF Electrical and Electronic Engineering
Bowen University, Iwo Nigeria.
isaac.ojedokun@bowen.edu.ng

Precious D. Agburuga[3]
Department OF Electrical and Electronic Engineering
Federal University Otuoke, Bayelsa State, Nigeria
agburugapd@fuotuoke.edu.ng

***Abstract*—The rapid proliferation of mobile devices has made Android-based systems a primary target for sophisticated cyber threats. Traditional intrusion detection systems often struggle with the class imbalance problem, where the overwhelming volume of benign traffic overshadows the critical signatures of malicious activity. This research proposes a robust binary classification framework for malicious Android traffic using static network behavior profiling and the Random Forest ensemble learning approach. Utilizing the CSE-CIC-IDS2018 dataset, we evaluated the model's performance in two distinct scenarios: an imbalanced environment and a balanced environment achieved through Random Under-Sampling (RUS). Initial results on the imbalanced dataset yielded an overall accuracy of 97.2%, yet demonstrated a significant deficiency in threat detection with a malicious recall of only 0.88. Upon applying the RUS technique, the model's performance improved substantially, achieving an Overall Accuracy of 98.7% and a Malicious Recall of 0.99. These findings indicate that while standard ensemble methods are effective, addressing data distribution through sampling techniques is essential to minimize False Negatives in cybersecurity applications. The study concludes that the integration of static profiling with RUS-balanced ensemble learning provides a highly reliable and computationally efficient solution for identifying Android-based network threats.**

Keywords—Intrusion Detection System (IDS); Random Forest (RF); Ensemble Learning; Android Network Security; Binary Classification; Static Feature Extraction ; Behavioral Profiling; Data Imbalance Mitigation

1. Introduction

The global dominance of the Android operating system has made it a primary focal point for cyber adversaries, ranging from individual hackers to organized state-sponsored groups [1]. Unlike traditional desktop environments, Android devices are characterized by a highly fragmented ecosystem of hardware and software, often resulting in delayed security patches and a vast attack surface [2,3]. As these devices become central to both personal and corporate data processing, handling everything from financial transactions to sensitive enterprise communications, the traditional perimeter-based security model has become insufficient [4,5]. Malicious actors increasingly exploit the mobile platform's constant connectivity to deploy botnets, facilitate data exfiltration, and conduct Distributed Denial of Service (DDoS) attacks, necessitating a move toward intelligent, network-centric defense mechanisms [6,7].

Traditional Intrusion Detection Systems (IDS) primarily relied on signature-based detection, which identifies known threats by matching specific byte sequences in network traffic [8,9]. However, the modern shift toward end-to-end encryption (TLS/SSL) and the emergence of polymorphic malware have rendered signature-based methods largely obsolete, as they cannot inspect encrypted payloads [10]. This

has catalyzed a shift toward flow-based analysis, where detection is based on the statistical and structural characteristics of communication, such as flow duration, packet inter-arrival times, and byte distributions [11,12]. To process this high-dimensional, complex data, machine learning algorithms, particularly ensemble methods like Random Forest, have emerged as the gold standard. These models can discern subtle behavioral "fingerprints" of malicious activity that are invisible to manual inspection or simple rule-based systems [13].

A significant challenge in developing effective machine learning-based IDS is the availability of high-quality, representative data [14]. For years, researchers relied on legacy datasets that failed to capture the complexity of modern network protocols and adversarial tactics. The CSE-CIC-IDS2018 dataset, developed by the Canadian Institute for Cybersecurity, addresses this gap by providing a massive-scale reflection of contemporary network traffic [15]. It includes a diverse range of attack scenarios and benign background noise captured in a controlled yet realistic environment. By applying ensemble learning techniques to this dataset, researchers can develop models that are not only accurate in a laboratory setting but also robust enough to handle the imbalanced and volatile nature of real-world Android network traffic.

2. Methodology

Developing a robust Intrusion Detection System (IDS) requires a meticulous pipeline that transforms raw network packets into actionable intelligence. This work details a six-phase, structured approach to building a robust binary classification model (Benign vs. Malicious Android Traffic) using the CSE-CIC-IDS2018 dataset and Random Forest Ensemble Learning.

2.1. Data Collection and Dataset Acquisition

The foundation of this work rests on the CSE-CIC-IDS2018 dataset, which represents a significant evolution in publicly available IDS benchmarks [17]. Unlike older datasets that rely on outdated traffic patterns, this dataset captures a massive scale of network events using a diverse infrastructure consisting of 450 nodes and 30 servers. It encompasses seven different attack scenarios, including Brute-force, DoS, DDoS, and Heartbleed, captured across distinct days of network activity. For this specific methodology, the focus is narrowed to identifying malicious Android-originated traffic through binary classification (Benign vs. Malicious).

The acquisition process involves extracting the raw PCAP files and the pre-processed CSV versions provided by the CIC. These files contain flow-based features generated by the CICFlowMeter-V3 tool, which aggregates packets into flows defined by the 5-tuple (source IP, destination IP, source port, destination port, and protocol). By utilizing this dataset, the study ensures that the model is trained on realistic, modern network behaviors that include both standard user activities and sophisticated adversarial tactics.

2.2 Data Balancing

In its raw form, the CSE-CIC-IDS2018 dataset is heavily skewed toward benign traffic. While this reflects real-world network conditions (where the vast majority of packets are legitimate), it poses a significant risk for ensemble learners like Random Forest. If trained on the imbalanced set, the model might develop a "majority class bias," where it achieves high accuracy simply by failing to identify rare but critical attack signatures.

Class imbalance was addressed through Random Under-sampling (RUS) of the benign class, aligning the number of normal traffic instances with the 2.7 million malicious samples. This method enables the Random Forest algorithm to learn from both classes equally. Enhanced sensitivity to minority patterns results from this, thus improving the overall F1-Score. The binary class distribution for the imbalanced versus balanced CSE-CIC-IDS2018 dataset is presented in Table 1. The binary class instance counts for the CSE-CIC-IDS2018 dataset shows the difference between the raw, imbalanced CSE-CIC-IDS2018 dataset and a balanced configuration. These figures are based on the full 10-day traffic capture, involving approximately 16.2 million total records

Table 1 Binary Class Distribution for the Imbalanced versus Balanced **CSE-CIC-IDS2018** dataset

Class Type	Imbalanced Mode (Original)	Balanced Mode (1:1 Ratio)
Benign (Normal)	13,484,708 (approx. 83.07%)	2,748,235
Malicious (Attack)	2,748,235 (approx. 16.93%)	2,748,235
Total Instances	16,232,943	5,496,470

2.3. Static Feature Extraction and Behavior Profiling

Static network behavior profiling focuses on the structural and statistical properties of communication flows rather than the volatile payload content, which is often encrypted in modern Android traffic. In this phase, we analyze the metadata of the network flows to identify "fingerprints" of malicious activity. This involves examining over 80 features provided in the

dataset, such as flow duration, packet length variance, inter-arrival times (IAT), and flag counts (e.g., SYN, ACK, PSH). These features serve as the primary indicators of how an Android device interacts with external servers.

Profiling also involves categorizing behavior based on flow directionality and volume. For instance, malicious traffic often exhibits unique statistical distributions—such as high-frequency small packet bursts during a DoS attack or unusual destination port patterns during a port scan. By profiling these static behaviors, the model can distinguish between the rhythmic, predictable patterns of legitimate background synchronization and the aggressive or irregular patterns associated with botnets or data exfiltration.

2.4. Data Preprocessing and Normalization

Raw network data is inherently "noisy" and often contains inconsistencies that can degrade model performance. The first step in this phase is Data Cleaning, which involves handling missing values (NaN) and infinite values (Inf) that often occur in calculations like "Flow Packets/s." These rows are either imputed using mean/median values or removed entirely if they represent a statistically insignificant portion of the data. Furthermore, duplicate entries and irrelevant metadata (such as Flow IDs or Timestamps) are stripped to prevent the model from "memorizing" specific events rather than learning generalizable patterns.

Following cleaning, Normalization is applied to ensure that features with large numerical ranges (like Byte counts) do not disproportionately influence the model compared to features with small ranges (like TCP flags). We typically employ Min-Max Scaling or Z-score Standardization:

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (1)$$

This mathematical transformation centers the data, allowing the Random Forest algorithm to converge more effectively and ensuring that the variance in network flow durations is treated with the same mathematical weight as the frequency of packet headers.

2.5. Feature Selection and Reduction

The high dimensionality of the CSE-CIC-IDS2018 dataset (80+ features) can lead to the "curse of dimensionality," where the model becomes overly complex and prone to overfitting. In this phase, we apply feature selection techniques to identify the most discriminative variables for detecting malicious traffic. We utilize Information Gain (Entropy) and Correlation Matrices to eliminate redundant features. For example, if "Total Fwd Packets" and "Subflow Fwd Packets" are perfectly correlated, one can be removed without losing predictive power.

Additionally, we leverage the Feature Importance scores inherent to the Random Forest algorithm itself. By training an initial "shallow" forest, we can rank features based on their Gini Impurity reduction. Features that contribute little to the decision-making process are pruned. This reduction not only improves the accuracy of the binary classification but also significantly decreases the computational overhead, making the model more suitable for real-time deployment on resource-constrained Android gateway environments.

2.6. Random Forest Ensemble Learning Model Development

The core of the detection engine is the Random Forest (RF) regressor/classifier, an ensemble learning method that constructs a multitude of decision trees during training. For binary classification, the RF algorithm uses "Bagging" (Bootstrap Aggregating) to train individual trees on different subsets of the data. This approach is particularly effective for IDS because it reduces the variance of the model, making it highly resilient to the outliers and noise typically found in network traffic. Each tree in the forest casts a "vote," and the majority class (Benign or Malicious) is selected as the final output.

During development, we tune several critical hyperparameters to optimize performance. This includes the number of trees ($n\_estimators$), the maximum depth of each tree (max_depth), and the minimum samples required to split an internal node. By utilizing a Grid Search with K-Fold Cross-Validation, we ensure that the model parameters are not just tuned to a specific slice of the CSE-CIC-IDS2018 data, but are robust enough to handle unseen traffic patterns. The resulting ensemble model benefits from the collective intelligence of hundreds of specialized trees, providing a much higher accuracy than any single decision tree could achieve.

2.7. Model Evaluation and Performance Analysis

The final phase involves a rigorous quantitative assessment of the model using a dedicated "test" subset of the data that was never seen during the training phase. We evaluate the model using a Confusion Matrix, which allows us to calculate four primary metrics: Accuracy, Precision, Recall (Sensitivity), and the F1-Score. In the context of intrusion detection, Recall is particularly vital, as it measures the model's ability to detect all malicious instances; a "False Negative" in this scenario could mean a compromised network.

Beyond basic metrics, we analyze the Receiver Operating Characteristic (ROC) Curve and calculate the Area Under the Curve (AUC). An AUC score close to 1.0 indicates that the Random Forest model has an excellent ability to distinguish between benign and malicious Android traffic. Finally, we measure the computational latency—specifically the time taken to classify a single flow—to ensure the methodology

meets the requirements for high-speed network monitoring. This comprehensive analysis confirms whether the static profiling and ensemble approach provide a viable defense mechanism against modern cyber threats. The summary of the core performance indicators used are presented in Table 2

Table 2 The summary of the core performance indicators used

Metric	Mathematical Formula	Description	Significance in IDS
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	The ratio of correctly predicted observations to the total observations.	Provides a general measure of how often the model is correct across both classes.
Precision	$\frac{TP}{TP+FP}$	The ratio of correctly predicted positive observations to the total predicted positives.	High precision indicates a low False Positive rate (fewer benign flows flagged as threats).
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	The ratio of correctly predicted positive observations to all observations in the actual class.	Crucial for security; measures the model's ability to catch all actual malicious attacks.
F1-Score	$2\square\left(\frac{Precision \cdot Recall}{\text{Precision+Recall}}\right)$	The weighted average of Precision and Recall.	The best measure for imbalanced datasets, balancing the trade-off between FP and FN.

3. Results and discussion

The performance evaluation results of the Random Forest model are presented in Table 3 Table 4 and Table 5. The results demonstrate the impact of dataset balancing on the performance of a Random Forest ensemble model when detecting malicious Android traffic.

3.1 Performance on Imbalanced Data (Table 3)

In the initial experiment using the imbalanced CSE-CIC-IDS2018 dataset, the model shows a clear bias toward the majority (Benign) class. While the Overall Accuracy is high at 97.2%, this figure is somewhat misleading due to the class distribution. The model achieves a near-perfect Recall of 0.99 for benign traffic, meaning it rarely misclassifies safe traffic as malicious.

However, the performance on the minority (Malicious) class reveals the challenges of imbalance. The Recall for malicious traffic drops to 0.88, indicating that 12% of actual attacks are going undetected. While the Precision remains respectable at 0.93, the lower F1-Score (0.90) for the malicious class suggests that the model struggles to generalize well for the specific patterns of mobile threats when they are overwhelmed by benign data samples.

3.2 Impact of Random Under-Sampling (Table 4)

The second experiment utilized Random Under-Sampling (RUS) to balance the dataset, leading to a significant and uniform improvement in detection capabilities. The Overall Accuracy increased to 98.7%, but more importantly, the disparity between class performance vanished.

The model's detection reliability saw a significant boost, with recall for the malicious class rising from 0.88 to 0.99, demonstrating a much higher effectiveness in capturing threats that were previously missed. Despite a reduction in majority class samples, precision for malicious traffic improved to 0.98, indicating that the model is making more accurate, confident classifications rather than simply guessing more often.Finally, the model achieved balanced harmony, with the macro average for precision, recall, and F1-score all reaching a consistent 0.98, proving high reliability and equal importance placed on both classes.

Table 3: Performance on Imbalanced CSE-CIC-IDS2018 Dataset

Metric	Benign (Majority)	Malicious (Minority)	Weighted Average
Precision	0.98	0.93	0.97
Recall	0.99	0.88	0.97
F1-Score	0.98	0.90	0.97
Overall Accuracy	**97.2%**		

Table 4: Performance on RUS Balanced CSE-CIC-IDS2018 Dataset

Metric	Benign	Malicious	Macro Average
Precision	0.99	0.98	0.98
Recall	0.98	0.99	0.98
F1-Score	0.98	0.98	0.98
Overall Accuracy	**98.7%**		

3.3 Comparative Summary

The shift from weighted averages (in Table 3) to macro averages (in Table 4) highlights a transition from a model that is "mostly right" because of the abundance of benign data, to a model that is "inherently robust." The result conclude that applying RUS effectively mitigates the "Accuracy Paradox" often found in cybersecurity datasets. By balancing the classes, the Random Forest ensemble was able to learn the distinct static network behavior profiles of malicious traffic more effectively, resulting in a 1.5% boost in accuracy and an 11% boost in attack detection (Recall).

Table 5 Comparative summary of the model's performance

Feature	Imbalanced Results	RUS-Balanced Results
Detection of Attacks (Recall)	88%	**99%**
False Alarm Rate (Benign Recall)	1%	2%
Reliability (F1-Score)	0.90 (Malicious)	**0.98 (Malicious)**
Overall Accuracy	97.20%	**98.70%**

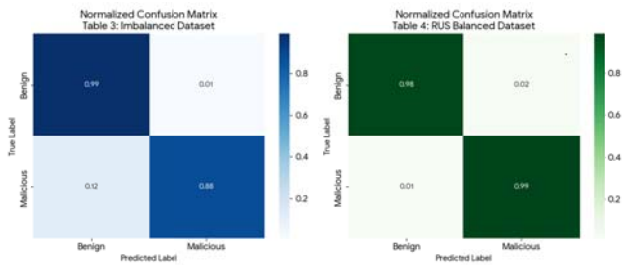

Figure 1 The confusion matrix for the results in Table 3 and Table 4

Table 6 Summary of Confusion Matrix Values

Metric (Normalized)	Imbalanced (Table 3)	RUS-Balanced (Table 4)
True Negative (Benign correctly identified)	0.99	0.98
False Positive (Benign labeled Malicious)	0.01	0.02
False Negative (Malicious labeled Benign)	**0.12**	**0.01**
True Positive (Malicious correctly identified)	**0.88**	**0.99**

The confusion matrices in Figure 1 and Table 6 illustrate the performance of the Random Forest model on both the imbalanced and balanced versions of the CSE-CIC-IDS2018 dataset. These diagrams are normalized to show the proportion of classifications relative to each true class, directly reflecting the Recall values reported in the research paper. The ion matrices highlight a significant shift in the model's error profile after applying Random Under-Sampling (RUS).

3.3.1 The Imbalanced Dataset confusion matrix

The model excels at identifying benign traffic, maintaining a high True Negative rate and a very low False Alarm rate of only 1%. However, due to its prioritization of the majority class in an imbalanced environment, it suffers from a 12% False Negative rate, where malicious activity "leaks" through and is incorrectly classified as safe. While this ensures minimal disruption to standard users, the resulting 0.88 recall for malicious traffic remains a key area for improvement

3.3.2 The RUS Balanced Dataset confusion matrix

RUS Balanced Dataset (Right Diagram) After balancing, the model demonstrates a dramatic improvement in threat detection, boasting a high uniform accuracy of 0.98 - 0.99 and increasing the malicious traffic True Positive rate (Recall) to 0.99. This enhancement significantly reduces the False Negative rate, plunging from 12% to only 1%, thereby minimizing the risk of undetected malicious traffic. Consequently, the slight, acceptable increase in the False Positive rate from 0.01 to 0.02 is a negligible trade-off for substantially higher security coverage

4. Conclusion

The development of a robust Intrusion Detection System (IDS) for Android network traffic requires a sophisticated balance between high-speed data processing and granular behavioral analysis. This work successfully demonstrated a structured, six-

phase methodology leveraging the CSE-CIC-IDS2018 dataset and a Random Forest ensemble learning approach. The findings of this work underscore the critical importance of addressing class imbalance when applying machine learning to Android malware detection. While the initial Random Forest ensemble model achieved a high overall accuracy of 97.2% on the imbalanced CSE-CIC-IDS2018 dataset, a granular analysis revealed a significant deficiency in threat detection, with a 12% False Negative rate. This "accuracy paradox" highlights how a model can appear successful by simply mastering the majority (benign) class while failing to adequately identify the high-risk minority (malicious) samples.

Again, by implementing Random Under-Sampling (RUS), this study demonstrated a substantial shift in model reliability. The balanced approach not only improved the Overall Accuracy to 98.7% but, more importantly, harmonized the detection capabilities across both classes. The recall for malicious traffic saw a dramatic increase from 0.88 to 0.99, effectively reducing the risk of undetected malware to just 1%. Ultimately, the combination of static network behavior profiling and a balanced ensemble learning strategy provides a robust framework for mobile security. The results confirm that RUS is a highly effective, computationally efficient preprocessing step that enables Random Forest models to capture the subtle signatures of malicious Android traffic without being overshadowed by the volume of benign data. This approach offers a more dependable solution for real-time intrusion detection systems where the cost of a missed attack far outweighs the minor increase in false positives.

References

1. AZUBUIKE, C. F. (2023). Cyber security and international conflicts: an analysis of state-sponsored cyber attacks. *Nnamdi Azikiwe Journal of Political Science*, *8*(3), 101-114.
2. Akhtar, Z. B. (2024). Operating systems (OS): An insight investigative research analysis and future directions. *Journal of Technology and Informatics (JoTI)*, *6*(1), 58-69.
3. Muhammad, Z., Anwar, Z., Javed, A. R., Saleem, B., Abbas, S., & Gadekallu, T. R. (2023). Smartphone security and privacy: A survey on APTs, sensor-based attacks, side-channel attacks, Google play attacks, and defenses. *Technologies*, *11*(3), 76.
4. Javier, M. (2021). Advancing Enterprise Connectivity with Zero Trust Network Access (ZTNA): Security Beyond the Perimeter. *International Journal of Trend in Scientific Research and Development*, *5*(2), 1324-1331.
5. Owobu, W. O., Abieba, O. A., Gbenle, P., Onoja, J. P., Daraojimba, A. I., Adepoju, A. H., & Ubamadu, B. C. (2021). Review of enterprise communication security architectures for improving confidentiality, integrity, and availability in digital workflows. *IRE Journals*, *5*(5), 370-372.
6. Anisetti, M., Ardagna, C., Cremonini, M., Damiani, E., Sessa, J., & Costa, L. (2020). Security threat landscape. *White Paper Security Threats*.
7. Abdi, A. H., Audah, L., Salh, A., Alhartomi, M. A., Rasheed, H., Ahmed, S., & Tahir, A. (2024). Security control and data planes of SDN: A comprehensive review of traditional, AI, and MTD approaches to security solutions. *IEEE Access*, *12*, 69941-69980.
8. Iyer, K. I. (2021). From signatures to behavior: Evolving strategies for next-generation intrusion detection. *European Journal of Advances in Engineering and Technology*, *8*(6), 165-171.
9. Nawaal, B., Haider, U., Khan, I. U., & Fayaz, M. (2024). Signature-based intrusion detection system for IoT. In *Cyber security for next-generation computing technologies* (pp. 141-158). CRC Press.
10. Jeebodh, M. R., & Baliyan, N. (2024, June). Iot malware detection using deep learning. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.
11. Giryes, R., Shafir, L., & Wool, A. (2024). A flow is a stream of packets: A stream-structured data approach for ddos detection. *arXiv preprint arXiv:2405.07232*.
12. Pekar, A., Makara, L. Á., Seah, W. K., & Rendon, O. M. C. (2023). Balancing Information Preservation and Data Volume Reduction: Adaptive Flow Aggregation in Flow Metering Systems. *Infocommunications Journal*, *15*(3), 82-94.
13. Maynard, L. (2025). *Advanced Machine Learning and Low-Dimensionality Projection Techniques for Enhanced GNSS Interference and Spoofing Detection* (Doctoral dissertation, University of Colorado Colorado Springs).
14. Thakkar, A., & Lohiya, R. (2023). A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System: A. Thakkar, R. Lohiya. *Archives of Computational Methods in Engineering*, *30*(7), 4245-4269.
15. Chinnasamy, R., Subramanian, M., Easwaramoorthy, S. V., & Cho, J. (2025). Deep learning-driven methods for network-based intrusion detection systems: A systematic review. *ICT Express*, *11*(1), 181-215.
16. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, *25*(15), 9731-9763.
17. Hewapathirana, I. U. (2025). A comparative study of two-stage intrusion detection using modern machine learning approaches on the CSE-CIC-IDS2018 dataset. *Knowledge*, *5*(1), 6.