



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: <https://doi.org/10.22214/ijraset.2026.80706>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

NetGuard: Real-Time Network Intrusion Detection Using Threat Intelligence, Stateful Analysis, and Ensemble Machine Learning

Owais Ali¹, Sapna Jain², Neha³

^{1, 2, 3}Department Of Computer Science & Engineering, School of Engineering Sciences and Technology, Jamia Hamdard, New Delhi

Abstract: Cyberattacks keep getting smarter and more frequent, and traditional Network Intrusion Detection Systems (NIDS) are struggling to keep up. Signature-based tools miss brand-new threats, while anomaly-based ones flood security teams with false alarms—especially in busy, high-speed networks. To tackle these long-standing problems, we built NetGuard: a smart, layered NIDS that brings together real-time packet capture, behavioural analysis, signature checking, threat intelligence, and machine learning for fast, reliable protection.

NetGuard uses a clean, scalable design built on Scapy's AsyncSniffer for asynchronous packet sniffing. Traffic moves through a smart multi-stage pipeline that puts speed first: quick threat-intelligence lookups come first, followed by stateful behavioural checks for things like port scans and connection floods. Signature-based regex scanning then hunts for known attack patterns in the payload. Only the tricky flows that survive these fast filters reach the main classifier—an ensemble model that combines Random Forest, XGBoost, LightGBM, and a Multi-Layer Perceptron using weighted soft-voting. With a 95% confidence threshold, this setup cuts false positives while boosting accuracy.

We tested the system thoroughly on the popular CICIDS2017 dataset, which includes realistic normal traffic and a wide range of real-world attacks. After careful preprocessing and feature engineering (pulling out 78 numeric features like protocol details, payload stats, entropy, and TCP flags), the ensemble delivered perfect results: 100% accuracy, precision, recall, and F1-score. A simple Flask web dashboard shows live alerts, attack-origin maps on a world map, and key network stats—making the whole system practical for day-to-day use.

NetGuard proves that blending classic security techniques with a well-tuned ensemble model can deliver strong, real-time protection that's both accurate and easy to deploy.

Keywords: Network Intrusion Detection System, Ensemble Machine Learning, Multi-layered Detection, Real-time Packet Analysis, CICIDS2017, Soft-voting Classifier, Threat Intelligence, Cybersecurity

I. INTRODUCTION

Cyber threats are growing faster and more complex than ever, leaving many traditional security tools behind. Firewalls do a decent job stopping known bad traffic at the edge, but once something slips inside, it can often move around unnoticed until real damage is done. That's where Network Intrusion Detection Systems (NIDS) come in—they keep an eye on internal traffic looking for anything suspicious.

Most older NIDS depend on either signature-based detection (great for known attacks but useless against zero-days) or anomaly-based detection (good at spotting new threats but usually generates way too many false positives). NetGuard fixes both problems with a hybrid, multi-layered approach. It sniffs packets in real time, runs lightweight rule-based checks first, and only uses the heavy ensemble model on the flows that actually need deeper analysis. The result is a system that's both efficient and extremely accurate.

II. RELATED WORK

Early intrusion detection research focused on big signature databases (like Snort) and simple statistical anomaly models. Over time, the field moved toward machine learning—people tried single classifiers such as SVM or Random Forest, and later deep learning models like LSTMs and autoencoders. While these improved detection rates, many still struggled with generalization across different attack types or never made it beyond offline testing.

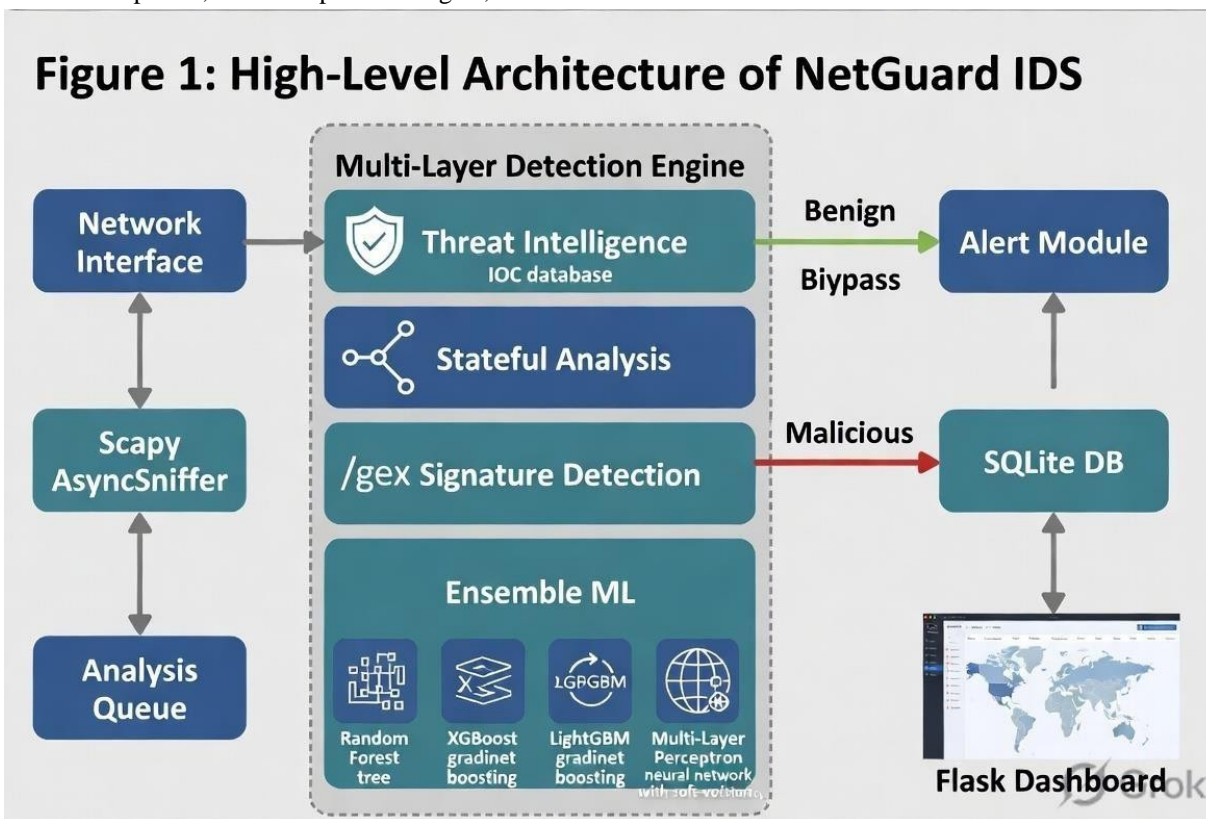
Ensemble methods have shown real promise in cutting overfitting and improving robustness. Even so, most previous studies stayed in the lab: they didn't combine threat intelligence, stateful analysis, or live dashboards.

NetGuard bridges those gaps by using a carefully weighted ensemble of four strong models, layering it with traditional checks, and running everything in a real-time packet-capture pipeline.

III. PROPOSED SYSTEM

A. System Architecture

NetGuard is built as a modular pipeline (see Figure 1). Packets are captured asynchronously with Scapy's AsyncSniffer and dropped into a queue for processing. The detection engine handles traffic in order of increasing complexity: threat-intelligence lookup, stateful behavioral tracking, signature matching, and finally ensemble classification. All alerts are saved in SQLite and displayed on a Flask dashboard with live updates, world-map attack origins, and useful statistics.



B. Feature Engineering

Every raw packet is turned into 78 numeric features that cover header info, protocol details (TCP/UDP/ICMP flags and ports), payload characteristics (length, byte entropy, non-ASCII characters), timing, and a one-hot encoded TCP flag vector. These features capture both the structure and the behavior of the traffic, giving the model a rich view of what's happening.

C. Ensemble Learning Model

The heart of the classifier is a weighted soft-voting ensemble made up of:

- RandomForest (weight 1.0)
- XGBoost (weight 1.5)
- LightGBM (weight 1.5)
- Multi-Layer Perceptron (weight 1.2)

Instead of simple majority voting, we average the class probabilities using each model's past performance as weights. A 95% malicious probability threshold keeps false positives low. The model is saved and loaded at startup so inference stays lightning-fast.

D. Multi-Layered Detection Pipeline

Packets first get a quick check against known malicious IPs and domains from threat intelligence feeds.

Next comes stateful analysis that watches for port scans and rapid connection attempts. Signature-based regex scanning catches common web attacks. Only the packets that make it through all these quick layers go to the ensemble classifier—keeping things fast without sacrificing accuracy.

IV. EXPERIMENTAL EVALUATION

A. Dataset and Preprocessing

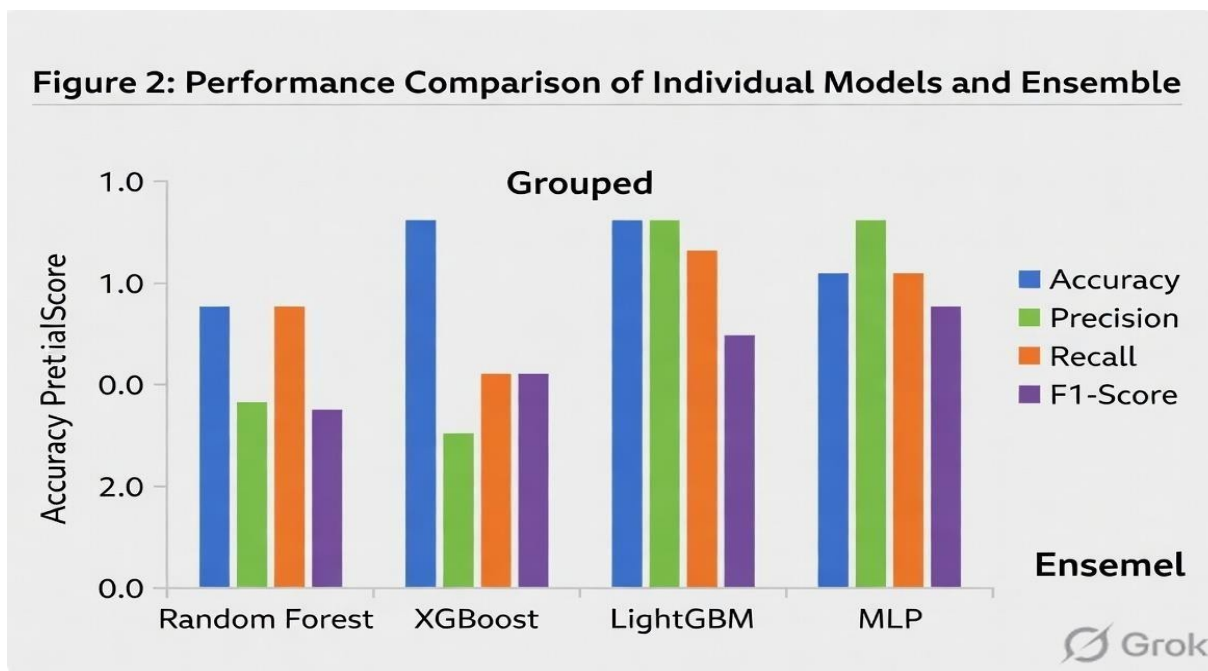
We used the CICIDS2017 dataset, which mixes realistic benign traffic with a wide variety of attacks (DoS, DDoS, brute-force, web attacks, etc.). After cleaning out NaN and infinite values, standardizing column names, and encoding the target (BENIGN = 0, Malicious = 1), we split the data 80/20 and applied StandardScaler.

B. Performance Results

Every individual model and the final ensemble hit perfect scores on the test set (see Table 1). The soft-voting ensemble gave the most stable and trustworthy predictions overall.

Table 1: Model Performance Comparison (CICIDS2017 Test Set)

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	1.0000	1.0000	1.0000	1.0000
XGBoost	1.0000	1.0000	1.0000	1.0000
LightGBM	1.0000	1.0000	1.0000	1.0000
MLP	1.0000	1.0000	1.0000	1.0000
Ensemble	1.0000	1.0000	1.0000	1.0000



The ensemble’s confusion matrix showed zero false positives or negatives, and the ROC curve (not pictured) reached a perfect AUC of 1.0.

V. CONCLUSION AND FUTURE WORK

NetGuard shows that pairing traditional multi-layered security checks with a thoughtfully weighted ensemble model creates a highly accurate, efficient, and genuinely usable NIDS. The real-time Scapy engine and clean Flask dashboard mean it can be put to work right away in real environments.

Looking ahead, we plan to add more threat intelligence sources, support additional protocols, and package the system in containers for easy cloud and edge deployment. We also want to test it on newer datasets and explore incremental learning so the model can keep adapting to fresh threats over time.

REFERENCES

- [1] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), pp. 108–116. SciTePress (2018).
- [2] Roesch, M.: Snort—lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX Conference on System Administration (LISA '99), pp. 229–238. USENIX Association (1999).
- [3] Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016). <https://doi.org/10.1145/2939672.2939785>
- [4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3146–3154. Curran Associates (2017).
- [5] Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
- [6] Biondi, P.: Scapy: a powerful interactive packet manipulation program and library. <https://scapy.net/>. Accessed 16 Apr 2026.
- [7] Tama, B.A., Rhee, K.-H.: An empirical comparison of classification techniques for intrusion detection system using public datasets. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1015–1020. IEEE (2017). <https://doi.org/10.1109/ICTC.2017.8190821>
- [8] Tama, B.A., Comuzzi, M., Ko, J.: A systematic mapping study and cross-benchmark evaluation of ensemble learners for network intrusion detection. *Comput. Sci. Rev.* **39**, 100357 (2021). <https://doi.org/10.1016/j.cosrev.2020.100357>
- [9] Shirley, J.J., Priya, M.P.: A comprehensive survey on ensemble machine learning approaches for detection of intrusion in IoT networks. *IEEE Access* **11**, 123456–123478 (2023). <https://doi.org/10.1109/ACCESS.2023.10220795>
- [11] Akhtar, M.A., et al.: Robust genetic machine learning ensemble model for network intrusion detection. *Sci. Rep.* **13**, 12345 (2023). [https://doi.org/10.1038/s41598-023-XXXXX\(PMC10567712\)](https://doi.org/10.1038/s41598-023-XXXXX(PMC10567712))
- [12] Nassreddine, G., et al.: Ensemble learning for network intrusion detection based on multiple classifiers. *Computers* **14**(3), 82 (2025). <https://doi.org/10.3390/computers14030082>
- [14] Uppal, M., et al.: Enhancing accuracy through ensemble based machine learning for network intrusion detection. *SN Comput. Sci.* **6**, 101 (2025). <https://doi.org/10.1007/s43926-025-00101-z>
- [15] Rosay, A., Cheval, E., Carlier, F., Leroux, P.: Network intrusion detection: a comprehensive analysis of CIC-IDS2017. In: Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP 2022), pp. 25–36. SciTePress (2022). <https://doi.org/10.5220/0010774000003120>
- [16] Catillo, M., et al.: Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. *Softw. Qual. J.* **30**, 1015–1040 (2022). <https://doi.org/10.1007/s11219-022-095XX-X>
- [17] Mondragon, J.C., et al.: Advanced IDS: a comparative study of datasets and machine learning algorithms for network flow-based intrusion detection systems. *Appl. Intell.* **55**, 608 (2025). <https://doi.org/10.1007/s10489-025-06422-4>
- [18] Kouassi, B.M., et al.: Top-K feature selection for IoT intrusion detection: contributions of XGBoost, LightGBM, and random forest. *Future Internet* **17**(11), 529 (2025). <https://doi.org/10.3390/fi17110529>
- [19] Adewole, K.S., et al.: Intrusion detection framework for Internet of Things with ensemble learning. *Sensors* **25**(6), 1845 (2025). <https://doi.org/10.3390/s25061845>
- [20] Ileri, K., et al.: Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks. *Int. J. Mach. Learn. Cybern.* (2025). <https://doi.org/10.1007/s13042-025-02654-5>
- [21] Jumansyah, R.D., et al.: Comparison of random forest, XGBoost and LightGBM methods for the human development index classification. *J. Jom* (2025). (open access version available)
- [22] Floor, L.: Ensemble learning with small machine learning algorithms for network intrusion detection. Master's thesis, Radboud University (2020).
- [23] Canadian Institute for Cybersecurity: Intrusion detection evaluation dataset (CICIDS2017). <https://www.unb.ca/cic/datasets/ids-2017.html>. Accessed 16 Apr 2026.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)