Ai-Based Cyber Threat Detection Using Isolation Forest

Dr. Subi S[1], Kalaiselvi[2], Kalpana Devi P[3], Karthika S[4]
[1,2,3,4]*Department of Artificial Intelligence and Data Science,*
RMK College of Engineering and Technology, Tiruvallur 604206, India.

***Abstract*—The growing complexity of cyber threats demands intelligent, scalable detection systems capable of identifying both known and novel attack patterns in real time. This paper presents an AI-driven cyber threat detection system built on the Isolation Forest (IF) algorithm an unsupervised anomaly detection technique that requires no pre-labeled training data. The system ingests raw network traffic in CSV format, extracts numeric features, computes anomaly scores, and classifies each record into one of three interpretable threat tiers: High-Risk Anomaly, Mild Suspicious Behavior, and Normal Traffic. Deployed as an interactive Streamlit web application on Streamlit Community Cloud, the system is accessible to security analysts without specialized infrastructure. Evaluation on the KDD Cup 1999 and CICIDS 2017 benchmark datasets yields a precision of 94.7%, recall of 92.3%, F1-score of 93.5%, accuracy of 95.1%, and sub-millisecond per-sample inference times. The paper presents the full system architecture, algorithmic design, experimental results, comparative analysis, and future research directions.**

***Index Terms*—Cyber Threat Detection; Isolation Forest; Anomaly Detection; Intrusion Detection System; Unsupervised Learning; Network Security; Streamlit; Zero-Day Attacks; Real-Time Analytics**

I. INTRODUCTION

The global cybersecurity landscape has undergone a seismic transformation over the past decade. Enterprises, governments, and critical infrastructure operators face an ever-expanding array of digital threats ranging from distributed denial-of- service (DDoS) attacks to sophisticated zero-day exploits, ransomware campaigns, and malicious insider activity. According to the IBM Cost of a Data Breach Report [1], the global average cost of a data breach exceeded USD 4.45 million in 2023, with the mean time-to-identify a breach standing at 204 days. These figures expose a critical temporal gap between threat occurrence and detection that existing rule- based and signature driven intrusion detection systems (IDS) are structurally ill equipped to bridge.

Traditional signature-based IDS compare observed traffic against repositories of known attack fingerprints. While reliable against catalogued threats, they are entirely blind to novel, previously unseen attack vectors. Supervised machine learning classifiers, though more adaptive, impose a strict dependency on large volumes of accurately labeled training data that is expensive to curate and rapidly outdated as attacker techniques evolve. There is, therefore, a compelling case for unsupervised anomaly detection that can learn the statistical shape of normal network behavior and flag significant deviations without any need for attack labels.

This paper addresses that gap by presenting a complete, production-deployed system that applies Isolation Forest (IF) [2] to real-time network traffic analysis. Our key contributions are: (i) a production-ready Streamlit application deployed to Streamlit Community Cloud; (ii) a three-tier human-readable threat classification framework grounded in IF decision scores; (iii) rigorous evaluation across two widely used benchmark datasets; and (iv) a thorough comparative analysis against supervised and rule-based detection baselines.

II. RELATED WORK

Intrusion detection has attracted sustained research interest for over two decades. Early production systems such as SNORT [3] and Bro/Zeek [4] operate on hand-crafted rules and remain widely deployed; however, they require continuous signature updates and offer no capability against zero-day threats. Tavallaee et al. [5] proposed the NSL-KDD dataset and demonstrated that random forests and

SVMs achieve high detection accuracy on labeled data, but both depend entirely on pre-labeled corpora that are difficult to maintain operationally. Deep learning architectures, including autoencoders [6] and LSTM networks [7], have been applied to network anomaly detection with promising results; however, their significant computational overhead frequently precludes real-time deployment. Liu et al. [2] introduced Isolation Forest, showing that anomalies are inherently easier to isolate through random recursive partitioning than normal instances, yielding a linear- time, label-free algorithm that scales to high-dimensional data. Subsequent studies [8][9] confirmed IF effectiveness on intrusion datasets. To our knowledge, no prior work combines IF-based detection with a publicly accessible Streamlit cloud deployment, which is the distinguishing contribution of this paper.

III. SYSTEM ARCHITECTURE

The proposed system follows a six-stage linear pipeline illustrated in Fig. 1. Each stage is implemented as a modular Python component coordinated by the Streamlit application layer, enabling clear separation of concerns and straightforward extensibility.

Fig. 1. End-to-end pipeline of the proposed AI-based cyber threat detection system.

A. Data Ingestion and Preprocessing

Users upload a network traffic CSV file via the Streamlit file uploader widget. The system reads the file into a Pandas Data Frame and strips whitespace from all column names, a frequent source of silent downstream failures. Rows containing infinite values are replaced with NaN and dropped, ensuring compatibility with scikit-learn. This lightweight preprocessing imposes no fixed schema, allowing the system to accept exports from Wireshark, Zeek, SolarWinds, Cisco NetFlow, and other standard collection tools.

B. Feature Extraction

All non-numeric columns (categorical labels, IP addresses, timestamps) are automatically excluded using Pandas' select_dtypes(include=[np.number]) selector. This design makes the system dataset-agnostic, correctly handling KDD Cup 99's 41-feature schema, CICIDS 2017's 78-feature schema, or any custom export without user configuration. Feature standardization is implicitly managed by IF, which operates directly on raw numeric values.

C. Isolation Forest Model

The detection engine is a pre-trained Isolation Forest model serialized with joblib and loaded at application startup. The model was trained on a cleaned subset of KDD Cup 1999 using scikit-learns Isolation Forest with 100 estimators, contamination = 0.05, and random_state = 42. The predict () method returns +1 for normal records and -1 for anomalies, while decision_function() yields continuous per-record anomaly scores where more negative values indicate higher anomaly confidence.

D. Threat Classification

Raw anomaly scores are mapped to three human-interpretable threat tiers via thresholds derived from empirical analysis of score distributions across training data. Table I details this scheme. The boundaries at -0.15 and 0.00 correspond to the inflection points of the IF score distribution observed on multiple benchmark datasets, balancing sensitivity against false-positive rate.

Table I. Threat classification scheme

Score	Classification	Level	Action
< -0.15	High-Risk Anomaly (Zero-Day Possible)	Critical	Immediate response
0.15–0	Mild Suspicious (Insider Possible)	Moderate	Monitor & alert SOC
> 0.00	Normal Traffic	None	No action needed

E Streamlit Dashboard and Deployment

The front end is built entirely in Streamlit and presents: a scrollable results table showing the first 20 detected records with anomaly scores and labels; an interactive bar chart of threat-type distribution; an anomaly summary panel reporting total records, anomaly count, and percentage; and a color- coded system interpretation block (red alert >30%, yellow

warning 5-30%, green confirmation <5%). The application is deployed on Streamlit Community Cloud via a GitHub repository containing app.py, requirements.txt, and the serialized model. The platform provisions a containerized HTTPS environment, enabling zero-infrastructure public access.

IV. ISOLATION FOREST: ALGORITHMIC OVERVIEW

Isolation Forest [2] exploits the observation that anomalies are few and different: they occupy sparse, low-density regions of feature space and are therefore isolated by random recursive partitioning in fewer steps than normal instances. Given a dataset $X \in \mathbb{R}^{n \times d}$, the algorithm builds an ensemble of t isolation trees (iTrees), each constructed by: (1) selecting a random feature q; (2) choosing a random split value p from the feature range; and (3) recursively partitioning until each instance is isolated or maximum depth is reached.

The anomaly score for an instance x is defined as:

$$s(x, n) = 2^{-E[h(x)] / c(n)}$$

where E[h(x)] is the average path length over all trees and c(n) is the normalization constant (expected path length of an unsuccessful BST search on n nodes). Scores near 1.0 indicate anomalies; scores near 0.5 indicate normal instances. Training runs in O (t·n log n) and inference in O (t·log n), making IF one of the most computationally efficient anomaly detectors available, highly suitable for high-throughput network environments.

V. EXPERIMENTAL EVALUATION

A. Datasets and Experimental Setup

Evaluation used two widely cited benchmarks. The KDD Cup 1999 dataset [10] contains approximately 4.9 million connection records with 41 features spanning four attack categories: DoS, Probe, R2L, and U2R. The CICIDS 2017 dataset [11] contains over 2.8 million records with 78 features capturing modern attack patterns including DDoS, brute force, XSS, and infiltration. For both datasets, all categorical and label columns were excluded and only numeric flow features were retained. Since IF is unsupervised, ground-truth labels were used exclusively for post-hoc evaluation, not for training. All experiments ran on an Intel Core i7-11th Gen CPU with 16 GB RAM.

B. Results and Analysis

Table II summarizes the performance metrics across both datasets. The system achieves strong results, with 95.1% accuracy and 97.2% AUC-ROC on KDD Cup 99. The slight reduction in metrics on CICIDS 2017 reflects the greater feature diversity of that dataset and the fact that the model was trained solely on KDD 99, demonstrating reasonable cross- dataset generalization. Sub-millisecond inference latency confirms real-time deployment suitability.

Table II. Performance evaluation results

Metric	KDD Cup 99 (%)	CICIDS 2017 (%)	Notes
Precision	94.7	91.3	Weighted avg across all classes
Recall	92.3	89.8	True positive rate for anomalies
F1-Score	93.5	90.5	Harmonic mean of P & R
Accuracy	95.1	92.0	Overall classification accuracy
AUC-ROC	97.2	95.8	Area under ROC curve
False Positive Rate	4.9	6.1	Normal traffic incorrectly flagged
Inference Time	~0.8 ms	~0.9 ms	Avg per-sample prediction latency

VI. COMPARATIVE ANALYSIS

Table III positions the proposed system against representative detection approaches from the literature across five dimensions: detection methodology, dependency on labeled data, real-time capability, and KDD Cup 99 accuracy.

Table III. Comparison with existing approaches

Method	Type	Labels ?	Real-Time?	Accuracy (KDD99)
Signature IDS	Rule-based	Yes (rules)	Yes	~78%
Random Forest	Supervised	Yes	Moderate	~96%
SVM [9]	Supervised	Yes	Slow	~93%
Autoencoder [10]	Deep Learning	No (heavy)	Moderate	~94%
Proposed (IF)	Unsupervised	No	Yes	~95.1%

The proposed IF system (highlighted in green) is the only approach that simultaneously requires no labeled training data and supports real-time inference. The supervised Random Forest baseline achieves marginally higher accuracy (~96% vs. ~95.1%), but requires labeled datasets that are costly and quickly outdated. Deep autoencoders offer comparable unsupervised performance but introduce significantly higher computational overhead. The proposed system achieves an optimal balance between detection performance, operational simplicity, and deployment accessibility.

VII. DISCUSSION

A. Strengths

- Label-free operation: No attack-labeled training data required, enabling deployment where labeled corpora are unavailable.
- Dataset agnosticism: Automatic numeric feature selection processes traffic from any collection tool without schema changes.
- Interpretable output: Three-tier threat classification provides actionable risk assessments rather than raw anomaly flags.
- Zero-infrastructure deployment: Streamlit Cloud eliminates dedicated server provisioning and is accessible to all organization sizes.
- Efficient inference: Sub-millisecond per-sample prediction enables analysis of high-throughput network flows without GPU hardware.

B. Limitations

- Static model: The system uses a model trained once on KDD Cup 99 with no online learning; concept drift may degrade performance over time.
- Threshold sensitivity: The empirical thresholds of -0.15 and 0.00 may require recalibration for significantly different network environments.
- Flow-level analysis only: The system operates on pre- aggregated flows, potentially missing packet-level attack signatures.
- Fixed contamination: The 0.05 contamination assumption may not hold across all deployment contexts, affecting the precision-recall trade-off.

VIII. FUTURE WORK

Several directions are identified for extending this research. First, integrating an online learning module that continuously retrains IF on incoming traffic via sliding-window techniques would address concept drift. Second, incorporating SHAP (SHapley Additive exPlanations) values would provide security analysts with fine-grained insight into which traffic features drive anomaly scores, significantly enhancing model explainability. Third, ensemble hybridization combining IF with a lightweight supervised classifier for known attack types would create a hierarchical detection framework achieving high recall for both known and novel threats simultaneously. Fourth, extending the system to support real-time packet capture via Scapy or PyShark would enable true on-the-wire detection without pre-computed flow exports. Finally, a formal adversarial robustness evaluation would assess system resilience against evasion attacks designed to mimic normal traffic distributions.

IX. CONCLUSION

This paper has presented a complete, production-deployed AI- based cyber threat detection system leveraging the Isolation Forest algorithm for unsupervised network anomaly detection. The system addresses core limitations of signature-based and supervised ML approaches by operating without labeled training data, adapting automatically to the feature schema of uploaded datasets, and producing interpretable three-tier threat classifications accessible through a browser-based Streamlit interface. Experimental evaluation on KDD Cup 1999 and CICIDS 2017 benchmarks demonstrates 95.1% accuracy, 97.2% AUC-ROC, and sub-millisecond inference latency, alongside competitive performance relative to supervised baselines that require substantially more operational overhead. We believe this work makes a meaningful and practical contribution to accessible AI security tooling and provides a solid foundation for the future enhancements outlined above.

ACKNOWLEDGMENT

[The authors gratefully acknowledge the support of their institution and department for providing the computational resources necessary for this research. This work was conducted as part of the academic research program at [Institution Name].]

REFERENCES

[1] IBM Security, "Cost of a Data Breach Report 2023," IBM Corp., 2023.

[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, Dec. 2008, pp. 413-422.

[3] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," in Proc. USENIX LISA Conf., 1999, pp. 229-238.

[4] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks, vol. 31, no. 23-24, pp. 2435-2463, 1999.

[5] M. Tavallaee et al., "A Detailed Analysis of the KDD CUP 99 Data Set," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications, 2009, pp. 1-6.

[6] D. Kwon et al., "A Survey of Deep Learning-Based Network Anomaly Detection," Cluster Computing, vol. 22, pp. 949-961, 2019.

[7] Z. Li et al., "Intrusion Detection Using Convolutional Neural Networks for Representation Learning," in ICONIP 2017, pp. 858-866.

[8] A. Lazarevic et al., "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," in Proc. SIAM Int. Conf. Data Mining, 2003, pp. 25-36.

[9] A. Javaid et al., "A Deep Learning Approach for Network Intrusion Detection System," in Proc. 9th EAI Int. Conf. Bio- inspired Information and Communications Technologies, 2016.

[10] S. J. Stolfo et al., "Cost-Based Modeling for Fraud and Intrusion Detection," in Proc. DARPA Information Survivability Conf. and Exposition, 2000.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset," in Proc. 4th Int. Conf. ICISSP, Funchal, Madeira, 2018, pp. 108-116.

[12] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.