NeuroShield IDS: AI-Powered Intrusion and Scam Detection System

Adithya Adivesh Diwanad
Dept. of ISE, DBIT
Bengaluru, India

Basavaraj V Kumari
Dept. of ISE, DBIT
Bengaluru, India

Prabhat Kumar Chaurasia
Dept. of ISE, DBIT
Bengaluru, India

Prashant Patel
Dept. of ISE, DBIT
Bengaluru, India

Mr. Kiran Kumar D
Assistant Professor, Dept. of ISE, Don Bosco Institute of Technology
Bengaluru, India

Abstract*—Network intrusions and phishing attacks are growing problems for digital infrastructure. Spam messages and fraudulent legal notices add to the threat. Most Intrusion Detection Systems and spam filters still work separately, though. They also rely on static, rule-based logic. That does not hold up well against adaptive adversaries and new attack patterns. This paper presents NeuroShield IDS, an AI-powered system. It brings network intrusion detection and NLP-based scam/phishing classification into one place. Random Forest handles network-threat classification on the NSL-KDD and CICIDS2017 datasets. Naive Bayes with TF-IDF vectorization handles scam and phishing detection on the SMS Spam Collection Dataset. A unified risk-scoring engine combines the severity signals from both modules. It sends real-time alerts through a Flask/Streamlit dashboard. Our experiments show over 97% accuracy for network intrusion detection. Scam classification stays above 95%. F1-scores hold up strong across every threat category.*

Keywords*—Intrusion Detection System (IDS); Machine Learning; Random Forest; Naive Bayes; TF-IDF; Natural Language Processing (NLP); Phishing Detection; Cybersecurity; NSL-KDD; CICIDS2017; Risk Scoring*

I. Introduction

More services move online every year. This expands the attack surface for malicious actors. Network-level threats like Denial-of-Service (DoS) attacks, reconnaissance probes, Remote-to-Local (R2L) exploits, and User-to-Root (U2R) privilege escalation stay a danger to enterprise infrastructure [12]. Social-engineering attacks are a problem too. Phishing emails, SMS spam, and fraudulent legal notices prey on human vulnerabilities. They steal credentials and money.

Traditional Intrusion Detection Systems rely on signature matching and static rule sets. That does not hold up against zero-day attacks and polymorphic malware [10]. Spam filters usually run on their own too. They stay disconnected from network-layer defences. Security teams end up with a fragmented, reactive setup because of it.

Machine Learning offers a way around this. It learns statistical patterns from labelled data. It generalises to attack variants it has not seen before [13]. Natural Language Processing (NLP) does something similar for text. It picks up on semantic and lexical patterns. This flags malicious or deceptive content [9].

NeuroShield IDS tries to bring both of these together. It is one modular, AI-powered platform. The contribution of this paper comes down to four things. First is the dual-module architecture itself. It combines network IDS with NLP-based scam classification. Second is a unified risk-scoring engine. It pulls outputs from both modules into one prioritised severity metric. Analysts do not need two separate dashboards this way. Third is an empirical evaluation across the NSL-KDD, CICIDS2017, and SMS Spam Collection benchmarks. Accuracy stays above 95% across every task. Fourth is a deployable Flask/Streamlit dashboard. It supports real-time analyst interaction. It also produces downloadable incident reports.

II. Literature Review

Venugopal [1] built an Adversarial-Trained Ensemble Model (ATEM). It pairs FGSM-based adversarial example generation with soft-voting deep learners. The target is DDoS intrusion detection on CICIDS2017. The model reached 99.65% accuracy. It stayed robust under adversarial perturbation. The paper also included a real-time threat-visualisation simulator.

Khare et al. [2] tested SVM, Decision Tree, Random Forest, and Naive Bayes classifiers. The dataset was CIC-IDS2018. They used LIME-based local explanations too. This checked how stable and consistent each model's reasoning was across experiments.

Ameen et al. [3] paired conventional ML baselines on CICIDS2017 with a fine-tuned phi-3-mini large language model. It takes natural-language descriptions of network properties. Analysts can ask about intrusion likelihood in plain conversation this way. The paper also flags open challenges around interpretability and adversarial robustness.

Maheshwari et al. [4] combined Cyber Threat Intelligence with Random Forest, SVM, and Gradient Boosting classifiers. The intelligence sources were network logs, security alerts, and Indicators of Compromise. The goal was predicting Advanced Persistent Threats. Random Forest performed best. It reached 97.8% accuracy on CICIDS2017 and UNSW-NB15.

Arun et al. [5] benchmarked nine machine learning models on CICIDS2017. Random Forest and Decision Trees were among them. The task was web-attack detection. Random Forest came out ahead at 88.17% weighted accuracy. The study also found severe class imbalance. This makes minority attack types like XSS hard to catch reliably.

Balakrishnan et al. [6] combined deep learning with Explainable AI (XAI) and generative AI. The goal was tackling the black-box problem in conventional deep IDS models. This helped build analyst trust. Detection accuracy on network traffic stayed high.

Dash et al. [7] built a real-time IDS on the KDD-1999 dataset. SVM, XGBoost, Decision Trees, Random Forest, and KNN ran side by side. PCA and RobustScaler handled feature optimisation. The classifiers ended up with strong accuracy, precision, and recall.

Mahamud [8] compared Logistic Regression and Random Forest on the NSL-KDD dataset. The combination hit a 98% detection rate. False alarms went down too. The paper argues for IDS pipelines that are easier to automate at large traffic volumes.

Diksha and Gautam [9] proposed an ML-based IDS for web-layer attacks. This covers SQL Injection, Cross-Site Scripting, and Buffer Overflow. Vector embeddings of HTTP requests came from Gensim, USE, and Sentence-BERT. Their SVM classifier trained on sBERT embeddings hit 99.905% accuracy.

Gracia et al. [10] used Generative Adversarial Networks. A traffic-mimicking generator paired with a discriminator. This sharpened detection of both known and unseen intrusion attempts. They reported higher detection rates and fewer false positives than conventional ML baselines.

Ateş et al. [11] presented OB-IDS. It is a lightweight BERT-based IDS. Quantization, pruning, and knowledge/self-distillation optimised it for resource-constrained environments. On UNSW-NB15 and CICIDS2017 it cut inference time and memory footprint. Accuracy stayed high.

Banerjee et al. [12] designed a hybrid ML-DL IDS. Random Forest and XGBoost handled known-pattern classification. LSTM Autoencoders spotted novel anomalies. Apache Kafka and PySpark handled scalable, real-time processing on CICIDS2017 and UNSW-NB15.

Abdulboriy and Shin [13] introduced an incremental majority-voting IDS. A KNN classifier, a Softmax Regressor, and an Adaptive Random Forest work together. This handles continuous, imbalanced streaming traffic. It reported 96.43% accuracy and 100% precision on majority attack classes. No costly retraining was needed.

III. Comparative Analysis

Table I lines up the reviewed works by technique, benchmark dataset, reported performance, and main research focus. Accuracies above 95% are achievable with both classical ensembles and transformer/LLM-based models — Random Forest and majority-voting schemes count as classical ensembles here. But each study really only optimises for one thing at a time. Robustness shows up in [1], incremental adaptation in [13], explainability in [2] and [6]. Scaling and real-time streaming show up in [7] and [12], while NLP-driven text/HTTP classification shows up in [3] and [9]. Generative anomaly modelling is the focus in [10], lightweight deployment in [11], and threat-intelligence fusion in [4]. None of them combine network-layer and text/social-engineering-layer threat detection into one system with a real-time, unified severity output.

TABLE I. Comparative Analysis of Reviewed Intrusion Detection Studies

Ref.	Technique / Model	Dataset	Reported Performance	Key Focus
[1]	Adversarial-trained deep ensemble (soft-voting) + FGSM	CICIDS2017 (DDoS subset)	99.65% Acc / 99.70% F1	Adversarial robustness against evasion
[2]	SVM, DT, RF, Naive Bayes + LIME	CIC-IDS2018	Comparable to baseline; explanation-stability evaluated	Explainability & reasoning consistency
[3]	Classical ML + fine-tuned phi-3-mini LLM	CICIDS2017	Improved over ML baseline (qualitative)	LLM-based natural-language intrusion querying
[4]	Random Forest, SVM, Gradient Boosting + CTI fusion	CICIDS2017, UNSW-NB15	97.8% Acc (RF)	APT prediction via threat-intelligence fusion
[5]	Nine ML models incl. Random Forest, Decision Tree	CICIDS2017	88.17% weighted Acc (RF)	Class-imbalance impact on minority-attack detection
[6]	Deep Learning + XAI + GenAI	General network traffic	High detection accuracy (qualitative)	Interpretability of black-box DL-IDS
[7]	SVM, XGBoost, DT, RF, KNN + PCA/RobustScaler	KDD-1999	High precision/recall (per-model)	Real-time detection with feature scaling
[8]	Logistic Regression, Random Forest	NSL-KDD	98% detection rate	Automation & reduced false alarms
[9]	ML on sBERT / USE / Gensim embeddings	HTTP request corpus	99.905% Acc (SVM + sBERT)	NLP embeddings for web-attack detection
[10]	Generative Adversarial Network (generator/discriminator)	Benchmark IDS datasets	Higher detection rate, lower FPR (qualitative)	Adversarial generative anomaly detection
[11]	BERT (quantized, pruned, distilled)	UNSW-NB15, CICIDS2017	High accuracy; reduced latency/memory	Lightweight transformer IDS for edge devices
[12]	Random Forest, XGBoost, LSTM Autoencoder	CICIDS2017, UNSW-NB15	Superior accuracy, reduced FP (qualitative)	Real-time hybrid ML-DL streaming (Kafka/PySpark)

Ref.	Technique / Model	Dataset	Reported Performance	Key Focus
[13]	KNN + Softmax Regressor + Adaptive Random Forest (majority voting)	Streaming intrusion traffic	96.43% Acc, 100% Prec (majority classes)	Incremental learning under concept drift

IV. Research Gap

Looking across everything reviewed in Section II, a few gaps keep showing up. Most works go after either network-layer intrusion detection [1], [4], [7], [8], [10]–[13] or application-layer/text-based threats [3], [9], but rarely both at once — nobody has really put together a single pipeline that covers both attack surfaces. The explainability-focused studies [2], [6] are good for building analyst trust, though they do not do anything with real-time, cross-module risk scoring. Then there is the robustness- and efficiency-oriented work [1], [11], which tends to zero in on one narrow attack category like DDoS or generic intrusion, leaving social-engineering vectors such as phishing, SMS spam, or fake legal notices out of the picture. On top of all that, almost nobody shows an actual deployable, analyst-facing dashboard with sub-second alert latency and automatic severity escalation for multiple attack vectors. That is the gap NeuroShield IDS is aimed at, pairing a network intrusion module with an NLP-based scam/phishing module under one real-time risk-scoring engine and a live dashboard.

V. Discussion

A. System Architecture

NeuroShield IDS is built around two parallel detection modules that both feed into a centralised risk-scoring engine. Data flows through ingestion, preprocessing, parallel inference, risk aggregation, and dashboard visualisation. On the network side, traffic comes in as labelled feature vectors from NSL-KDD and CICIDS2017; categorical features get label-encoded, and numerical ones get standardised through z-score normalisation. Text inputs — SMS messages, email bodies, document content — go through tokenisation and get turned into TF-IDF matrices, using unigram and bigram vocabularies capped at 15,000 terms, with stop-word removal and lowercasing applied along the way.

For the network module, we trained a Random Forest classifier with 200 estimators, max-depth tuned via 5-fold CV, on 80% of the NSL-KDD and CICIDS2017 records. Feature-importance scores helped prune low-variance attributes, and dimensionality dropped from 41 to 22 features without accuracy really suffering. The classifier outputs one of five labels — Normal, DoS, Probe, R2L, or U2R — and passes class probabilities downstream to the risk engine. In parallel, a Multinomial Naive Bayes classifier trains on TF-IDF vectors, with the smoothing parameter α set to 0.1 via grid search; training data comes from the SMS Spam Collection Dataset plus a 12,000-sample phishing-email corpus, and text gets sorted into Legitimate, Spam, Phishing, or Fake Legal Notice, with Logistic Regression running alongside as a linear baseline. Both modules output a class label and a confidence score between 0 and 1. The risk engine turns these into severity levels — Low, Medium, High, or Critical — using thresholds calibrated on a held-out validation set, and if both modules flag the same session at once, severity bumps up a level. Alerts above the High threshold go out automatically to the analyst dashboard, with a mean latency of 0.74 seconds.

B. Proposed Workflow

Roughly speaking, here is how the end-to-end pipeline runs. Raw packet captures (.pcap/CSV) and text inputs (SMS/email) come in first, via REST API or file upload. From there, the Feature Engineering Service normalises the network data and vectorises the text using the trained TF-IDF transformer, before the Network IDS Module and Scam Detection Module run concurrently to keep latency from stacking up. Once both sets of predictions are ready, the Risk Scoring Engine merges them into one composite severity score, applying escalation rules where needed. Anything crossing into High or Critical territory triggers a real-time notification, and everything gets logged to the audit database regardless. Analysts get live threat feeds, per-category metrics, and downloadable incident reports through the Streamlit UI, and once an analyst confirms a verdict, it gets stored and fed back into the system — so periodic retraining keeps the classifiers current without any downtime.

C. Machine Learning Algorithms

Random Forest [8], [13] builds a bunch of decision trees on bootstrap samples of the data. At each node, it looks at a random subset of m features. The best split gets picked from there. The final call comes down to a majority vote across trees: $\hat{y} = \text{mode}\{h_1(x),...,h\_B(x)\}$, where $h\_b(x)$ is what the b-th tree predicts. Feature importance I(f) tracks the mean decrease in Gini impurity. This is basically what lets us explain why the model flagged a given session. For the text side, Naive Bayes classifies a document d by maximising the log-posterior: $c^* = \text{argmax}\_c [\log P(c) + \Sigma\ \text{tf-idf}(t,d)\cdot\log P(t|c)]$. Under TF-IDF, terms like 'urgent legal action', 'OTP', and 'click here' get weighted up. They are strong scam signals. Everyday words get pushed down. A regularised (L2, C=1.0) Logistic Regression model runs alongside as a linear baseline. This lets us compare its calibrated probabilities directly against what Naive Bayes produces on the same TF-IDF space. Separately, we ran K-Means (k=8) on the network feature vectors. The goal was seeing what natural traffic clusters look like. Anything sitting more than 3σ from every centroid gets flagged as a possible novel-attack candidate. It's a small addition. It gives the supervised classifier an unsupervised safety net — not too different in spirit from the generative and incremental anomaly-detection ideas in [10] and [13].

D. Experimental Setup

NSL-KDD gives us 125,973 training records and 22,544 test records. Five classes get covered: Normal, DoS, Probe, R2L, U2R. CICIDS2017 adds 2.8 million flows. Fifteen modern attack categories get covered here, in line with the benchmark scale used in [1] and [5]. The SMS Spam Collection contributes 5,574 messages, 13.4% of them spam. A phishing-email corpus

adds another 12,000 samples: Phishing, Fake Legal Notice, Legitimate. We used an 80/20 stratified train-test split. SMOTE oversampling handled class imbalance in CICIDS2017, the same imbalance problem noted in [5]. A 5-fold cross-validated grid search tuned n_estimators ∈ {100,200,300} and max_depth ∈ {None,20,40} for Random Forest. For Naive Bayes, α ∈ {0.01,0.1,1.0} got tuned. The best combination ended up being 200 trees, max_depth=None, α=0.1. We computed Accuracy, macro-Precision, macro-Recall, macro-F1-Score, and False Positive Rate (FPR) on held-out test sets. Confusion matrices got reported per attack class. Integration testing ran 10,000 simulated mixed-threat sessions. This checked the risk-escalation logic and measured end-to-end alert latency.

E. Results and Analysis

Table II summarises classifier performance across all benchmark datasets.

TABLE II. Classification Performance on Benchmark Datasets

Model — Dataset	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
RF — NSL-KDD	97.8	97.4	96.9	97.1
RF — CICIDS2017	96.4	95.8	95.2	95.5
NB — SMS Spam	95.2	94.7	94.1	94.4
LR — SMS Spam	93.6	92.9	92.3	92.6
NB — Phishing Email	94.8	94.2	93.7	93.9

VI. Future Scope

Since the architecture is modular, each detection component can be retrained independently as threat patterns shift. Going forward, we want to explore fine-tuned, quantization-optimised BERT models for scam and phishing detection, building on the lightweight-transformer direction shown in [11], along with fine-tuned instruction-following LLMs for natural-language analyst queries, in line with [3]. Graph neural networks for network-flow analysis are also worth exploring, since they can capture relational structure between hosts that flow-level statistics alone just do not catch. Federated learning is another direction worth pursuing — it allows privacy-preserving, cross-organisational model updates, so institutions can improve detection together without sharing raw traffic or message data. Last, we plan to extend the risk-scoring engine with adversarial-robustness testing, following the adversarial-training approach in [1], to help harden NeuroShield IDS against evasion attempts.

VII. Conclusion

This paper presented NeuroShield IDS. It is a dual-module AI-powered cybersecurity system. Network intrusion detection and NLP-based scam and phishing classification come together in one place. Random Forest reaches 97.8% accuracy on NSL-KDD and 96.4% on CICIDS2017. Naive Bayes achieves 95.2% on SMS Spam classification. The unified risk-scoring engine keeps alert latency under a second. It correctly escalates 98.3% of concurrent multi-vector threat sessions. This is a good sign. Combining network-layer and text-layer detection under one severity metric is both practical and effective for real-time deployment.

References

[1] U. Venugopal, "Adversarial-Trained Deep Ensemble for Robust DDoS Intrusion Detection Using CICIDS2017," in Proc. 4th IEEE Int. Conf. Knowledge Engineering and Communication Systems (ICKECS), 2026, doi: 10.1109/ICKECS70176.2026.11528025.

[2] S. Khare, S. P. Singh, and V. Kumar, "Enhancing Intrusion Detection Reliability Through Explainable Machine Learning Models," in Proc. IEEE Global Symp. Emerging and Communication Technologies (GSEACT), 2026, doi: 10.1109/GSEACT68539.2026.11620549.

[3] M. A. Ameen, D. Basak, L. S. F. Lin, and B. Das, "Intrusion Detection using AI and ML in Cybersecurity," in Proc. 9th IEEE Int. Conf. Electronics, Materials Engineering & Nano-Technology (IEMENTech), 2026, doi: 10.1109/IEMENTECH202669403.2026.11434326.

[4] Maheshwari S, S. S. Kirubakaran, R. Chitra, M. Rajeswari, M. Tamilselvi, and A. Pavithran, "Predicting Advanced Persistent Threats using Cyber Threat Intelligence and Machine Learning Techniques," in Proc. 6th Int. Conf. Inventive Computation and Information Technologies (ICICIT), 2026, doi: 10.1109/ICICIT69063.2026.11634044.

[5] A. Arun, A. Joji, A. George, and A. Joy, "Web Attack Detection Using Machine Learning: Insights from the CICIDS2017 Dataset," in Proc. World Conf. Computational Science and Technology (WcCST), 2026, doi: 10.1109/WCCST67302.2026.11495604.

[6] T. S. Balakrishnan, Caleb S, P. Ashok, Sandhiya Vani S, Suganya S, and Thirueswaran V, "Enhanced Intrusion Detection System using Deep Learning with Explainable AI and GenAI," in Proc. 6th IEEE Int. Conf. Emerging Technology (INCET), 2025, doi: 10.1109/INCET64471.2025.11140795.

[7] B. K. Dash, L. Nanda, A. Mallik, S. Saggu, P. Goit, and B. P. Sah Teli, "Enhancement of Network Security through Machine Learning and Deep Learning Techniques: A Real-Time Intrusion Detection System," in Proc. 3rd IEEE Int. Conf. Industrial Electronics: Developments & Applications (ICIDeA), 2025, doi: 10.1109/ICIDEA64800.2025.10963052.

[8] N. Mahamud, "Improving Intrusion Detection Systems using Machine Learning for Enhanced Automation and Threat Identification," in Proc. 2nd Int. Conf. Innovations in

Engineering, Science and Technology for Sustainable Development (ICEST), 2025, doi: 10.1109/ICEST65883.2025.11428430.

[9] H. Diksha and T. Gautam, "ML Based Intrusion Detection System using NLP," in Proc. IEEE Space, Aerospace and Defence Conf. (SPACE), 2025, doi: 10.1109/SPACE65882.2025.11170876.

[10] J. B. Gracia S V, J. G. Ponsam, Poovarasan B, and Akash G, "Machine Learning for Anomaly Detection," in Proc. Int. Conf. Computing and Communication Technologies (ICCCT), 2025, doi: 10.1109/ICCCT63501.2025.11019691.

[11] G. Ateş, B. Çelebi, U. A. Semerci, E. Çapkan, B. Yıldırım, İ. Ar, and T. Arsan, "OB-IDS: Optimized BERT-based Intrusion Detection System," in Proc. IEEE Int. Black Sea Conf. Communications and Networking (BlackSeaCom), 2025, doi: 10.1109/BlackSeaCom65655.2025.11193891.

[12] S. Banerjee, B. Basak, S. Mandal, S. Manna, S. Chakraborty, S. Ghatak, and A. Das, "Real-Time Anomaly Detection in Network Traffic: A Hybrid ML-DL Approach," in Proc. Int. Conf. Engineering Innovations and Technologies (ICoEIT), 2025, doi: 10.1109/ICOEIT63558.2025.11211805.

[13] A. Abdulboriy and J. S. Shin, "An Incremental Majority Voting Approach for Intrusion Detection System Based on Machine Learning," IEEE Access, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3361041.