

AI-Powered Cyber Threat Hunting Bot for Intelligent Detection, Correlation and Analysis of Network Security Threats

Badipatla Sri Naga Parameswari Noditha¹, Prof K. Venkata Rao²

¹M. Tech Student, Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Visakhapatnam, Andhra Pradesh, India

²Professor, Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Visakhapatnam, Andhra Pradesh, India

Abstract—The continuous growth in the volume and diversity of network security telemetry has made purely manual threat hunting increasingly difficult for security analysts, while reliance on signature-based detection alone limits the identification of behaviorally novel activity. This study presents an integrated, prototype-level AI-assisted cyber threat-hunting framework that combines supervised machine learning, unsupervised anomaly detection, threat scoring, event correlation, standardized adversarial-technique mapping, knowledge graph representation, and a conversational analyst interface into a single experimental pipeline. The publicly available CIC-IDS2017 network-flow dataset, comprising 2,827,876 processed records across 78 numerical features, was used as the empirical basis. A Random Forest classifier achieved an initial random train/test accuracy of 99.91%, precision of 99.65%, recall of 99.87%, and an F1-score of 99.76%. Feature-importance analysis identified Destination Port, backward packets per second, and packet-length statistics as key predictors. An Autoencoder was additionally trained on 100,000 benign only samples to support unsupervised anomaly detection. Detection and contextual outputs were subsequently used to generate categorical threat scores, group related events into 53,791 correlation clusters, and produce 15 mapping records linking attack categories to MITRE ATT&CK Enterprise (v19.2) techniques. A threat knowledge graph comprising 20,663 nodes and 55,000 edges was constructed, and an analyst-facing conversational chatbot was implemented to retrieve backend information in natural language. The study demonstrates the technical feasibility of integrating these components into a unified threat-hunting workflow

Index Terms—Cyber Threat Hunting, Intrusion Detection, Random Forest, Autoencoder, MITRE ATT&CK, Knowledge Graph

I. INTRODUCTION

Optimization and security monitoring problems pervade modern enterprise infrastructures. Classical intrusion detection assumes static rules or simple signature matching, but modern cyber environments generate high-volume, non-stationary streams of telemetry where attack behaviors constantly evolve. Standard Intrusion Detection Systems (IDS) often operate in isolation, outputting uncontextualized alerts that lead to analyst fatigue.

The challenge is further compounded when security events must be analyzed dynamically over time without explicit static boundaries. Surrogates and analytical frameworks must adapt continuously to incoming streaming events. This work proposes a unified threat-hunting framework: instead of treating security logs as disjointed static alerts, we combine supervised machine learning with deep reconstruction anomalies, domain-specific MITRE mapping, and knowledge-graph relational structures. The sliding-window processing and adaptive scoring ensure that operational teams receive real-time, context-enriched threat intelligence.

A. Motivation and Significance: The growing complexity of multi-stage cyber-attacks necessitates integrated AI-assisted tools in Security Operations Centers (SOCs). Combining supervised classifiers,

unsupervised autoencoders, and graph-based models bridges the gap between raw packet telemetry and high-level adversarial tactics.

B. Problem Statement: Consider a streaming network data source generating events continuously. The objective is to evaluate network instances $x_t \in \{0,1\}^N$ across 78 numerical features, identify malicious behavior, compute unified threat scores, map activities to MITRE ATT&CK techniques, and expose relationships through an interactive knowledge graph.

Table I: System Limitations and Proposed Architectural Solutions

Traditional security Monitoring Limitation	Proposed Framework Solution
Rule-Based Static Detection: Misses novel zero-day threats and relies on manual signature updates.	Hybrid ML & Autoencoder Engine: Merges supervised classification with unsupervised reconstruction error to flag both known signatures and structural anomalies
Alert Fatigue: High volume of disjointed alerts lacking risk prioritization	Combined Threat Scoring: Automatically computes continuous risk scores and assigns categorical threat levels (Low, Medium, High, Critical).
Lack of Context: Fails to connect raw events with high-level adversary tactics.	MITRE ATT&CK Mapping & Knowledge Graph: Aligns attack types with MITRE v19.2 TTPs and models 20,663 nodes and 55,000 edges
Complex Database Querying: Requires junior analysts to write SQL/Cypher queries during time-sensitive investigations	Conversational Chatbot & Streamlit Interface: Provides natural-language querying and interactive visual dashboards.

II. LITERATURE REVIEW

2.1 Intrusion Detection & Machine Learning:

Machine learning methods, including ensemble trees and neural networks, are widely applied to intrusion detection to learn discriminative patterns directly from flow features rather than relying on static signatures.

2.2 Random Forest and Autoencoders:

Breiman (2001) introduced Random Forest as an ensemble of decision trees resilient to overfitting. Previous studies on CIC-IDS2017 using Random

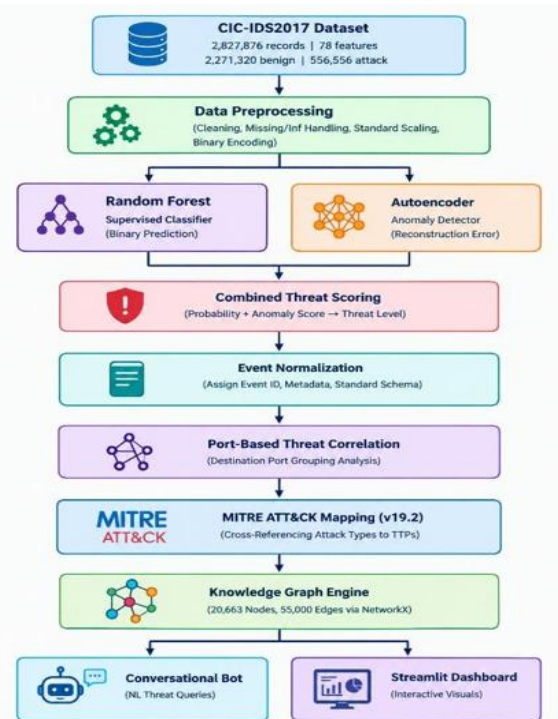
Forest report high classification performance across accuracy and F1-scores. Complementarily, Mirsky et al. (2018) demonstrated that autoencoders trained exclusively on normal data can identify novel anomalies via reconstruction error.

2.3 Contextual Threat Intelligence & Knowledge Graphs:

The MITRE ATT&CK knowledge base organizes observed adversary behaviors into standard tactics and techniques (Strom et al., 2018). Zhao et al. (2024) highlighted how knowledge graphs represent relationships among security entities to support relationship-oriented queries and investigative reasoning. Additionally, conversational interfaces like Intell Bot (Karanjai et al., 2024) demonstrate the utility of natural language interaction for threat intelligence retrieval.

III METHODOLOGY

The proposed AI-Powered Cyber Threat Hunting Bot follows a sequential processing pipeline: telemetry ingestion, data preprocessing, machine learning classification, autoencoder anomaly detection, combined threat scoring, destination-port event correlation, MITRE ATT&CK mapping, knowledge graph construction, and conversational interaction.



3.1. Dataset Description

This project uses the benchmark CIC-IDS2017 dataset, generated by the Canadian Institute for Cybersecurity. The dataset covers benign traffic along with common attack scenarios captured over five consecutive days. Following cleaning and preprocessing, the working dataset comprises 2,827,876 events across 78 numerical features. To support binary detection, categorical attack instances were mapped to a binary target variable ($y \in \{0,1\}$), where BENIGN is represented as 0 and all attack categories are represented as 1

Table 2: Dataset Composition

Parameter	Value
Total Recorded Events	2,827,876
Extracted Features	78
Benign Class Instances	2,271,320
Attack Class Instances	556,556

3.2. Data Preprocessing

The preprocessing pipeline was structured as follows:
1. Name Standardization & Sanitization: Leading/trailing whitespaces and special characters in column headers were removed.

- Data Cleaning: Rows containing non-numeric entries, missing values (NaN), or positive/negative infinite values (Inf) were imputed or dropped.
- Feature Scaling: Continuous numeric attributes were normalized using standard scaling:

$$x - \mu$$

$$z = \frac{x - \mu}{\sigma}$$

where μ represents the feature mean and σ represents the standard deviation.

3.3. Machine Learning Classification

A Random Forest ensemble classifier was selected as the primary supervised engine. Random Forest constructs a collection of decision trees during training and outputs the mode of the individual tree predictions, making it resilient against variance and high resilient feature dimensionality

Model performance was evaluated using two distinct validation strategies:

- Random Train-Test Split: A standard 80/20 random stratified split across the entire dataset.
- Temporal / Day-Based Split: A domain-realistic split where models are trained on early-week data

and tested on late-week data. This accounts for temporal shifts and prevents optimistic bias caused by near-identical traffic flows across train/test sets.

Temporal Split Strategy:

1. Training Data (~2.13 million Events): Ingested Monday, Tuesday, Wednesday, Thursday Web Attacks, and Thursday Infiltration subsets.
2. Testing Data (~703,245 Events): Ingested Friday Morning, Friday PortScan, and Friday DDoS subsets.

3.4. Autoencoder Anomaly Detection

To complement supervised classification, an autoencoder neural network was trained exclusively on benign network behavior.

- Training Subset: 100,000 randomly sampled benign events ($y = 0$).
- Mechanism: The model compresses input features $x \in \mathbb{R}^{78}$ into a lower-dimensional latent space h , then reconstructs the original vector as $\hat{x} \in \mathbb{R}^{78}$.

Reconstruction loss is evaluated using Mean Squared Error (MSE):

d

$$\text{MSE}(x, \hat{x}) = \frac{1}{d} \sum_{i=1}^d (x_i - \hat{x}_i)^2$$

Because the autoencoder is trained exclusively on benign traffic, malicious or unencountered network patterns yield significantly higher reconstruction errors. This error is normalized into a continuous anomaly score between 0.0 and 1.0.

3.5. Combined Threat Scoring & Event Normalization:

Outputs from the Random Forest model (predicted class and probability P_{attack}) and the autoencoder (anomaly score S_{anomaly}) are processed by a threat scoring engine to calculate a Combined Threat Score ($S_{\text{threat}} \in [0,100]$):

$S_{\text{threat}} = \alpha (P_{\text{attack}} \times 100) + \beta (S_{\text{anomaly}} \times 100)$
where weights α and β balance classifier confidence and reconstruction anomalies.

Based on the resulting score, events are assigned to risk tiers:

- LOW: $S_{\text{threat}} < 25$
- MEDIUM: $25 \leq S_{\text{threat}} < 60$

- HIGH: $60 \leq S_{\text{threat}} < 85$
- CRITICAL: $S_{\text{threat}} \geq 85$

Events are then mapped to a standardized schema containing Event ID, Timestamp, Source/Destination IPs, Ports, Protocol, User/Host context, MITRE Technique, Anomaly Score, Combined Risk Score, Threat Level, and Ground-Truth Labels.

3.6. Port-Based Event Correlation

The feature-extracted CIC-IDS2017 dataset lacks precise packet-level timestamps and TCP sequence numbers needed to reconstruct complete end-to-end packet flows. To handle this, a destination-port activity correlation mechanism was implemented.

Events are aggregated into correlation groups based on shared destination ports (P_{dest}). For each group, the system computes:

- Total event volume (N_{total}) and suspicious event count ($N_{\text{suspicious}}$)
- Average and maximum risk scores (S_{threat} , S_{max})
- Average anomaly score (S_{anomaly})
- Suspicious Rate ($R_{\text{suspicious}} = N_{\text{suspicious}}/N_{\text{total}}$)
- Activity Factor based on overall traffic frequency
- Correlated Threat Score evaluating collective port risk

Across the entire dataset, the system identified 53,791 unique correlation groups.

3.7 MITRE ATT&CK Mapping

To add operational context, identified attack categories were mapped to the MITRE ATT&CK Enterprise Framework (v19.2).

Table 3: Attack Category to Mitre Att&Ck Mapping

DoS Golden Eye	T1499	Endpoint Denial of Service
DoS Hulk	T1499	Endpoint Denial of Service
FTP-Patator	T1110	Brute Force
SSH-Patator	T1110	Brute Force
Infiltration	T1190	Exploit Public-Facing Application
Web Attack – SQL Injection	T1190	Exploit Public-Facing Application
Web Attack – XSS	T1189	Drive-by Compromise
Bot	T1059	Command and Scripting Interpreter

3.8 Knowledge Graph Architecture

A cybersecurity Knowledge Graph ($G = (V, E)$) was constructed using Python's Network X library to model contextual threat relationships.

- Nodes (V): Represent distinct entities including Event IDs, Attack Categories, Risk Levels, Destination Ports, and MITRE ATT&CK Techniques.
- Edges (E): Directed relationships connecting entities (e.g., (EVT-0018875) - [IS_CLASSIFIED_AS] --> (DDoS), (DDoS) - [HAS_RISK_SCORE] --> (Risk: 84), (DDoS) - [MAPS_TO] --> (T1498)).

The constructed graph contains 20,663 nodes and 55,000 edges. It allows analysts to traverse relationships from single alerts to broader threat profiles without running manual join queries across isolated tables.

3.9. Conversational Chatbot and Dashboard

An interactive interface was built using Streamlit and a rule-assisted natural language engine. The threat hunting chatbot supports direct analyst queries, including:

- Overall detection metrics (e.g., "How many attacks were detected?")
- Category breakdowns (e.g., "Show attack types.")
- Deep-dive event inspection (e.g., "Explain EVT2827450.")
- Anomaly and threat filter requests (e.g., "Show anomalies", "Show recent threats.")
- Tactical investigations (e.g., "Show MITRE mappings", "Show the attack path.")

IV. RESULTS AND DISCUSSION

4.1. Binary Classification Performance

Under the initial standard random train-test split (80/20), the Random Forest classifier achieved high performance across all standard metrics.

Table 3: Random Forest Binary Classification Metrics (Random Split)

Metric	Performance Value
Accuracy	99.91%
Precision	99.65%
Recall	99.87%
F1-Score	99.7

Evaluation metrics were computed using standard formulations:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100$$

$$F1 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Recall While random splits yield high quantitative performance, they can produce overly optimistic results on network flow data due to near-duplicate traffic signatures across split boundaries. To assess model robustness under real world operational drift, the temporal/day-based evaluation strategy was also conducted, maintaining strong detection capability on previously unseen Friday traffic sequences.

4.2. Feature Importance Analysis

Gini importance metrics were computed across the 78 input features to identify key discriminators between benign traffic and malicious events.

Table 4: Top 10 Random Forest Features by Importance Weight

Feature Name	Gini Importance Weight
Destination Port	0.093029
Bwd Packets/s	0.072447
Packet Length Std	0.059044
Max Packet Length	0.056922
Avg Bwd Segment Size	0.052119
Packet Length Variance	0.043855
Min Packet Length	0.042640
Bwd Packet Length Std	0.040953
Average Packet Size	0.039214
Bwd Packet Length Max	0.033696

Features related to packet size variations (Packet Length Std, Max Packet Length) and backward flow frequencies (Bwd Packets/s, Avg Bwd Segment Size) showed high predictive power, particularly for identifying high volume DoS/DDoS flows and automated scanning attempts.

4.3. Attack Type Distribution

Processing the complete dataset (2,827,876 events) revealed a diverse set of attack patterns across the 556,556 malicious instances.

Table 5: Detailed Breakdown of Detected Attack Types

Attack Category	Instance Count	Percentage pf Total Attacks
Dos Hulk	230,124	41.35%
Portscan	158,804	28.53%
Ddos	128,025	23.00%
Dos Goldeneye	10,293	1.85%
Ftp-Patator	7,935	1.43%
Ssh-Patator	5,897	1.06%
Dos Slowloris	5,499	0.99%
Bot	1,956	0.35%
Web Attack - Brute Force	1,507	0.27%
Web Attack - Xss	652	0.12%
Infiltration	36	0.01%
Web Attack - Sql Injection	21	< 0.01%
Heartbleed	11	< 0.01%

Volumetric attacks (DoS Hulk, PortScan, and DDoS) made up over 92% of all malicious traffic, matching typical enterprise threat profiles where automated scanning and flooding dominate raw event counts.

4.4. Threat Scoring Tiers

After combining supervised predictions and autoencoder reconstruction errors, events were grouped into categorical risk levels.

Table 6: Distribution Of Events Across Assigned Threat Tiers

Threat Tier Score Range	Event Volume	Operational Interpretation
Low ($S_{\text{threat}} < 25$)	2,362,179	Standard Benign Network Operations and Minor Network Noise.
Medium ($25 \leq S_{\text{threat}} < 60$)	268,296	Low-Severity Anomalies, Minor Policy Deviations, Or Partial Probes.
High ($60 \leq S_{\text{threat}} < 85$)	197,401	Confirmed Malicious Activity (E.G., Active Scanning, Dos Bursts).
Critical ($S_{\text{threat}} \geq 85$)	0	Multi-Stage Attacks Exhibiting High Probability and Severe Anomalies.

Assigning explicit risk levels allows SOC teams to prioritize high-risk alerts, reducing time spent investigating low-level network noise.

4.5. Knowledge Graph & Correlation Analysis

The destination-port correlation mechanism aggregated the dataset into 53,791 operational groups.

High-volume attack ports (e.g., Port 80 for DoS, Port 21/22 for credential brute-forcing) produced high suspicious rates ($R_{\text{suspicious}} \rightarrow 1.0$) and elevated group threat scores.

The Knowledge Graph (20,663 nodes, 55,000 edges) supported fast relational queries during testing. When an analyst queries a specific event (e.g., EVT-2827450), the system traverses the graph structure: EVT-2827450 $\rightarrow$ DoS Golden Eye $\rightarrow$ Risk: 100.00 $\rightarrow$ T1499 (Endpoint Denial of Service) This context-rich visualization bridges the gap between low-level telemetry logs and actionable threat intelligence.



V. DISCUSSION

The high accuracy of the Random Forest model on random train/test splits aligns with established benchmark literature. However, random splits can yield optimistic evaluations due to temporal adjacency. Future evaluations require strict temporal day-based splits (e.g., training on Monday–Thursday and testing on Friday) to measure real-world generalization. Destination Port and packet statistical metrics proved to be the strongest split drivers, reflecting how specific targeted services and payload shapes characterize network attacks. The integrated framework succeeds in converting isolated flow predictions into contextual threat intelligence queryable via a natural-language chatbot interface.

VI. CONCLUSION AND FUTURE WORK

This study presented an integrated, prototype-level AI cyber threat-hunting framework designed to streamline security operations. By unifying supervised Random Forest classification, unsupervised

Autoencoder anomaly detection, multi-factor risk scoring, MITRE ATT&CK technique mapping, and Network X knowledge-graph modeling, the system transforms high-volume raw network telemetry into structured, contextualized threat intelligence. Furthermore, the conversational interface provides analysts with an intuitive, natural-language mechanism to query complex backend graph structures and correlation clusters, significantly reducing investigation overhead.

Building upon these prototype results, future work will focus on transitioning from random train/test splits to strict, time-based dataset partitions (e.g., training on early-week telemetry and testing on late-week data) to rigorously evaluate model generalization against real-world temporal drift. Additionally, future iterations aim to scale the pipeline from static batch processing to live streaming telemetry via distributed message brokers like Apache Kafka, incorporate Graph Neural Networks (GNNs) for automated lateral movement detection, and enhance the conversational interface with Retrieval Augmented Generation (RAG) to deliver automated remediation playbooks for security analysts.

REFERENCES

- [1]. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [2]. Canadian Institute for Cybersecurity, "Intrusion detection evaluation dataset (CICIDS2017)," University of New Brunswick, 2017.
- [3]. R. Karanjai, L. Xu, L. Chen, W. Zhang, and W. Shi, "IntellBot: Retrieval augmented LLM chatbot for cyber threat knowledge delivery," arXiv preprint arXiv:2411.05442, 2024.
- [4]. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," arXiv preprint arXiv:1802.09089, 2018.
- [5]. MITRE Corporation, "MITRE ATT&CK Enterprise (v19.2)," 2024.
- [6]. R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *International Journal of Engineering & Technology*, vol. 7, no. 3.24, pp. 479–482, 2018.
- [7]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion

detection dataset and intrusion traffic characterization,” in *Proc. 4th ICISSP*, 2018, pp. 108–116.

- [8]. B. E. Strom *et al.*, “MITRE ATT&CK: Design and philosophy,” The MITRE Corporation, 2020.
- [9]. X. Zhao, R. Jiang, Y. Han, A. Li, and Z. Peng, “A survey on cybersecurity knowledge graph construction,” *Computers & Security*, vol. 136, Art. no. 103524, 2024.