Threat Hunting Assistant Using GenAI

Jhanviben Desai
Masters of CyberSecurity and Threat Intelligence
University of Guelph
Guelph, ON, Canada
jhanvibe@uoguelph.ca

Vidhi Parekh
Masters of CyberSecurity and Threat Intelligence
University of Guelph
Guelph, ON, Canada
parekhv@uoguelph.ca

***Abstract* - The growing size and complexity of cybersecurity telemetry have made manual threat hunting increasingly time-consuming and error-prone for security analysts. To address this challenge, this project presents a Generative Artificial Intelligence (AI)-powered Threat Hunting Assistant designed to automate the analysis and summarization of intrusion detection system (IDS) logs. The system combines machine learning-based anomaly detection with natural language summarization to help security teams efficiently interpret large volumes of network data. Publicly available datasets, specifically CICIDS2017, were used to simulate real-world attack scenarios, including port scans, brute-force attempts, and DDoS activity. The development pipeline involved data cleaning, feature selection, machine learning classification using K-Nearest Neighbors (KNN), and summarization with the T5-base transformer model. The T5 model generates concise, human-readable summaries of detected malicious events to reduce analyst workload and improve situational awareness. This proof-of-concept demonstrates the potential for AI-driven tools to enhance threat detection, streamline incident triage, and support scalable threat hunting operations within security teams.**

Index Terms—Threat Hunting, Generative AI, LLM, GPT-3.5, IDS Logs, Prompt Engineering, Log Summarization.

I. Introduction

Cybersecurity threats are growing rapidly in frequency, complexity, and impact-targeting critical systems, enterprise infrastructure, and connected devices across the globe. By the end of 2025, global financial losses due to cybercrime are projected to exceed $10.5 trillion USD annually, highlighting the urgent need for more intelligent and proactive defense mechanisms [1]. In response, national and international cybersecurity standards, such as the NIST Cybersecurity Framework have been widely adopted to guide the identification, detection, protection, response, and recovery processes involved in managing cyber risks [2].

One of the most vital elements of contemporary cybersecurity planning is Cyber Threat Hunting (CTH) — an active procedure where analysts examine telemetry data, logs, and threat intelligence to detect latent threats that are likely to evade conventional detection methods. Nonetheless, CTH deployment in operations is confronted with a number of challenges: security analysts are overwhelmed by having to comb through massive amounts of heterogeneous endpoint, network, and system logs, while current tools tend to produce too many false positives or too few contextual descriptions. The heightened sophistication of attacks and the accompanying log volumes have made threat hunting by hand ineffective, error-ridden, and unscalable.

To overcome these limitations, researchers and practitioners have experimented with machine learning (ML)-driven anomaly detection tools, which can potentially reveal both known security threats and unknown threats. These tools are, nevertheless, largely non-transparent, unexplainable, and non-real-time operable—properties that are key to operational trust and analyst decision-making [3]. The application of Explainable AI (XAI) frameworks within cybersecurity workflows has been a move in the right direction but, for all that, their deployment remains limited given usability concerns and integration hurdles [4].

To this end, Large Language Models (LLMs) like GPT-3.5 and Mistral-7B have been effective tools to enhance cyber threat hunting. The models are able to comprehend and summarize large volumes of complex log data, create human-readable intelligence, and deliver contextual reasoning through structured prompts. Their flexibility is particularly useful in operations that demand natural language understanding, synthesis, and interpret abilities of growing significance to cybersecurity operations.

In this paper, we present the prototype and design of a Threat Hunting Assistant based on Generative AI. Our tool leverages LLMs to summarize intrusion detection system (IDS) security logs, e.g., CICIDS2017, and produce actionable threat hypotheses. The assistant preprocesses and normalizes raw log data, uses prompt engineering methods to formulate security questions, and incorporates open-source threat intelligence (e.g., MITRE ATT&CK, MISP) to provide contextual awareness. Our solution offers automation and interpretability in contrast to traditional SIEM tools or offline ML models, allowing human analysts to rapidly comprehend sophisticated threat scenarios.

II. Motivation

The ever-expanding digital infrastructure has exponentially increased the attack surface of modern organizations. Cybersecurity teams now have to deal with a relentless barrage of threats—from zero-day attacks to targeted, persistent intrusions—against everything from large enterprises to government systems of supreme criticality. In spite of the pervasive use of sophisticated automated tools such as Security Information and Event Management (SIEM) systems and Intrusion Detection Systems (IDS), the onus of separating actual threats from daunting amounts of alerts still lies with human analysts. The difficulty herein stems from the volume, diversity, and intricacy of log data that is generated at network, system, and endpoint levels.

Manual threat hunting effectively detects stealthy or unknown threats but is limited by human constraints, making it time-consuming and error prone. Existing rule-based solutions and traditional machine learning models lack the flexibility to understand new attack

patterns, leading to high false positives and slow response times.

This research is driven by an urgent need to supplement human cyber analysts with intelligent, context-aware tools that can automatically summarize heterogeneous log data and produce actionable threat leads. We suggest creating an LLM-based Threat Hunting Assistant relying on models such as GPT-3.5 and Mistral-7B. Such models allow for parsing complicated security logs, semantic explanation of anomalous events, and producing concise, human-readable summaries and leads for investigation.

By incorporating threat intelligence feeds such as MITRE ATT&CK and MISP, and using prompt engineering methods specific to security activities, the assistant is designed to close the gap between raw telemetry and analyst discernment. Finally, this work tackles the burning issue of scalability and information overload in contemporary threat hunting workflows and offers a solution that provides improved efficiency, accuracy, and situational awareness in cybersecurity operations.

III. Methodology

This section describes the pipeline for the Threat Hunting Assistant using Generative AI. It includes data preparation, feature engineering, machine learning model training, and summarization with a transformer model.

Fig.1 flow diagram

A. Data Collection :

The information utilized for this project was gathered from the CICIDS2017 dataset, which is open sourced via Kaggle and the Canadian Institute for Cybersecurity[18]. The data mimics typical network traffic, with both benign and malicious activity accounted for in all network situations such as brute force attacks, DoS, PortScan, DDoS, and infiltration events. Each data point consists of over 70 features such as source/destination IPs, port numbers, flow duration, protocol, byte/packet counts, and flow behaviors.

B. Data Preprocessing :

Because of the nature of CSV files in CICIDS2017, there were a number of preprocessing tasks needed:

- Concatenation: All of the daily CSV log files were concatenated into one big DataFrame.
- The occurrence of missing and infinite values, namely NaN and Inf, was identified and replaced or removed to provide consistency.
- Attack-type labels such as DDoS, PortScan, and BruteForce were all merged under one label "Attack" to simplify binary classification (Attack vs Benign). Features that had a single distinct value were removed, considering their inability to discriminate.
- Features like source_file were removed as they were used for accounting and were not predictive.
- The final dataset contained only statistically and contextually relevant fields, free of any noise or bias [21].

C. Feature Selection and Analysis :

To enhance model performance and reduce dimensionality, several feature selection techniques were employed:

- A Pearson correlation matrix identified highly correlated features (e.g., with a correlation greater than 0.9) for potential exclusion.
- A RandomForestClassifier ranked features by importance. The top 20 features, including FlowBytes/s and Init_Win_bytes_forward, were selected.
- The ANOVA F-value identified attributes with the highest variance across classes.
- Features with a variance below 0.01 were removed to eliminate static or irrelevant columns.
- These methods produced a concise and relevant set of features for model training [22].

D. Model Training :

Features and labels were separated, with the binary target label column (Attack vs Benign) encoded using LabelEncoder from scikit-learn. Deduplication was applied to the dataset to avoid identical records influencing both training and testing phases. The dataset was split into 70% training and 30% testing, with stratified sampling to maintain class distribution. An overlap check confirmed no shared records between training and testing sets, eliminating the risk of data leakage. Feature scaling was applied using StandardScaler to standardize numerical features. K-Nearest Neighbour (KNN) with k=5 neighbors for distance-based classification. Model performance was evaluated using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

E. Automated Summarization Using Generative AI

To convert complex log entries and model predictions into readable summaries for SOC analysts:

- The T5-base transformer model from Hugging Face's Transformer library was selected for its efficient sequence-to-sequence capabilities [14].
- Key features such as flow duration, packet count, and protocols from each record were concatenated into a natural language sentence prefixed with "summarize:".
- T5 configuration:

  max_length: 200 tokens

  min_length: 20 tokens

  do_samples: False (to ensure deterministic outputs)
- The resulting summaries included relevant features and the prediction result, enabling analysts to quickly comprehend the context and severity of each log.

IV. Experimental settings

This section covers the setup, algorithms, evaluation methods, and infrastructure for network attack detection and summarization experiments.

A. Classification Models:

We initially considered three algorithms for structured tabular classifications:

- Random Forest: A robust ensemble decision tree model that ranks feature importance.
- K-Nearest Neighbors (KNN): A simple and effective distance-based classifier.
- Support Vector Machine (SVM): Evaluated but excluded due to long training times with the complete feature set.

Random Forest and KNN were ultimately selected for their performance and efficiency.

B. Dataset Partitioning and Validation:

The CICIDS2017 dataset was split 70/30 for training and testing, maintaining the distribution of "Attack" and "Benign" classes through stratified sampling. We used 5-fold cross-validation for model validation and applied StandardScaler for feature scaling. A deduplication check prevented data leakage between sets.

C. Evaluation Metrics:

We assessed model performance using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. These metrics provided insights into the models' classification abilities.

D. Generative AI Summarization

The T5-base Transformer model generated human-readable insights from network records. Each input included relevant features with "summarize:" to enhance performance. Configuration parameters included:

- Maximum Length: 50 tokens
- Minimum Length: 20 tokens
- Sampling: Disabled for consistent output

This setup resulted in clear summaries for SOC analysts.

E. Computational Resources

Experiments were conducted on a Lenovo Legion 5 Pro laptop with:

- Processor: AMD Ryzen™ 7 5800H (8 cores, 16 threads)
- Memory: 16 GB RAM
- GPU: NVIDIA® GeForce® RTX 3060

This hardware was adequate for training models and conducting summarization using the Hugging Face Transformers library.

V. Result

The original CICIDS2017 dataset contained 2,830,743 instances with 80 features. Following preprocessing steps—removal of missing/infinite values, constant columns, and non-informative metadata—the resulting dataset had 2,827,876 instances. Label normalization was also done to simplify classification by consolidating all malicious behavior under a single "Attack" label, reducing it to a binary classification problem (Attack or Benign).

In order to estimate feature importance, a Random Forest classifier was used for feature importance estimation. The top 20 features that contributed the most are presented in Fig. 2.

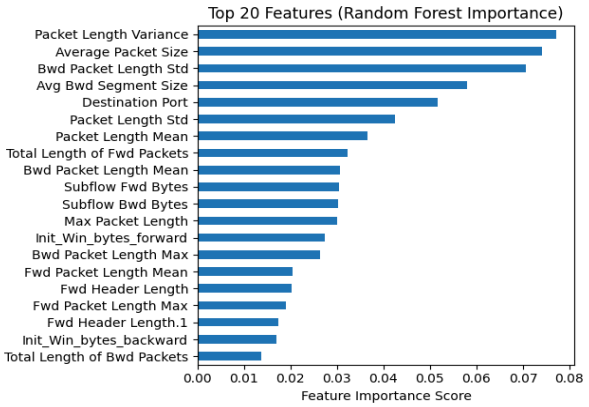

Fig.2 top 20 features from random forest importance

Among them, Packet Length Variance, Average Packet Size, Bwd Packet Length Std, and Destination Port were found to be important to separate attack and benign flows. The ranking guided both feature selection as well as natural language summarization generation.

Random Forest and K-Nearest Neighbors (KNN) models were employed in the classification model, while Support Vector Machine (SVM) was tried but omitted due to its high computational cost. Preprocessing comprised stratified 70/30 train-test splitting, deduplication, overlap checks to prevent data leakage, and numerical feature scaling with StandardScaler. Model training was conducted with cross-validation applying 5-fold cross-validation for generalizability.

The KNN classifier performed well. The confusion matrix and classification report in Fig. 3 illustrate good predictive power:

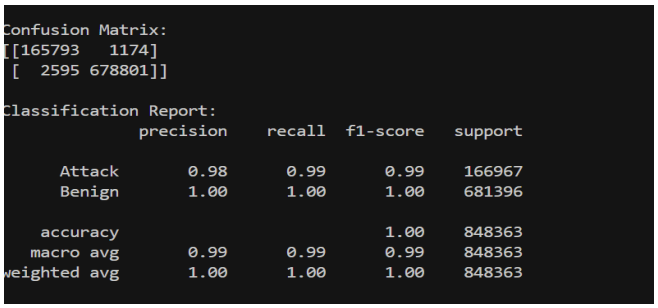

Fig.3 confusion matrix and classification report of knn

These findings corroborate the capacity of the classifier in discriminating between benign and malicious traffic almost perfectly. This is credited to efficient data cleaning, pertinent feature selection, and class-balancing while train-test splitting. Furthermore, the highly structured nature of the CICIDS2017 dataset enabled well-defined class separability.

In order to aid human interpretation, a summarization module was implemented with the T5-base transformer model. Inputs to summarize were formatted around significant attributes (e.g., packet lengths, ports, byte statistics) and converted into natural language prompts with the prefix "summarize:". The model produced brief summaries with a minimum of 20 and a maximum of 50 tokens and created coherent and pertinent insights.

For example, one of these outputs is shown in Fig.4:

Fig.4 summary generated from t5 base model

This brief overview provides analysts with a quick view of possible threat vectors without having to interpret raw logs.

By highlighting the most revealing features and exposing them in plain English, the assistant minimizes mental overhead and speeds up decision-making. For instance, successive summaries involving anomalous packet size variation or suspicious ports (e.g., 80, 8080, 4444) can instantly trigger a port scan or exfiltration analysis.

VI. Discussion

The creation of the Threat Hunting Assistant through the use of Generative AI illustrates the strength of a hybrid strategy, blending conventional machine learning techniques with natural language processing, in significantly improving cyber threat detection and management. The exemplary classification accuracy obtained with the K-Nearest Neighbors (KNN) model illustrates the merit of meticulous data preprocessing, judicious feature selection, and adequately balanced training procedures. The model was highly effective at network anomaly detection, with 0.98 precision and 0.99 recall for the "Attack" class, keeping both false positives and false negatives low.

Feature importance analysis verified the findings of the security community, revealing that features such as packet length variance, mean packet size, and destination port are major pointers to indicative malicious activity. Through statistical methods and tree-based methods (e.g., Random Forest Importance and the ANOVA F-test), the project was able to retain only the most important features while avoiding loss of information, thus generating an efficient model with ease of interpretation.

One of the most significant innovations in the project is the use of generative AI summarization with the T5-base model. By translating complex network records into easy-to-read, simplified summaries, the assistant bridges the gap between machine output and human decision-making. The summaries allow SOC analysts to identify threats quickly, contextualize behavior, and initiate response sequences without visual log inspection in tabular form.

At the end of the project, we recognized that a more diverse dataset could have further improved the results and overall robustness of the model. While the CTU-13 dataset was available, its primary focus is on botnet detection, making it less aligned with our project's objective of general attack classification. The ADFA IDS dataset was another option, but it is highly complex and would have required significant time for preprocessing and integration, which was beyond our project timeline. Although the dataset we used may not have been the most ideal, it was readily available, relevant to our objectives, and suitable for building and validating our models within the given time constraints.

We evaluated Random Forest, SVM, and K-Nearest Neighbour (KNN) for attack detection. Random Forest achieved 100% accuracy after eliminating overfitting and data leakage concerns, largely due to the cleaned, well-structured dataset. SVM, however, required excessive training time (3–5 hours) and delivered inconsistent results, making it impractical. KNN provided ~99% accuracy, excellent precision and recall for both attack and benign traffic, and required minimal computational resources. Given its strong performance, efficiency, and interpretability, KNN was chosen as the final classification model.

In the final stage of the project, we aimed to generate concise summaries of network flow data. Initially, we experimented with the GPT-3.5 model, as it offers the flexibility to set custom instructions and focus the summary on specific features. However, due to API rate limits and associated costs, GPT-3.5 was not a practical choice for our project. As an alternative, we opted for the T5-base model, which provided a free, offline solution for summarization. While GPT-3.5 likely offers superior output with more contextual control, T5-base delivered acceptable results for summarizing single records. Although basic summaries could also be generated using rule-based scripts, such static methods lack the variability, context awareness, and dynamic focus that transformer-based models like T5 or GPT can provide.

In the current proof-of-concept, malicious network logs are manually identified and passed to the summarization model. However, in a real-world deployment, this process would be automated. Network logs from sources such as firewalls, intrusion detection systems (IDS), or SIEM platforms would be ingested in real time through log pipelines or APIs. Once a log is flagged as malicious by the trained detection model (e.g., KNN), it would automatically be structured into a natural language prompt and routed to the summarization engine, powered by the T5-base model. The summarizer would generate concise, human-readable

outputs highlighting key threat attributes such as ports, packet characteristics, and anomaly patterns. These summaries would be delivered to SOC dashboards, incident response platforms, or automated alerting systems, providing immediate, actionable threat intelligence to security analysts without manual intervention.

VII. Conclusion

This project effectively proved the feasibility of the design and development of a Generative AI-based Threat Hunting Assistant for identifying and explaining network anomalies. With the integration of a traditional machine learning classifier and an LLM-based summarizer, the system provides both high-performance detection and human-readable explanations.

The nearly 100% accurate classifier, confirmed by confusion matrix metrics and cross-validation, was only leveraging the most influential features of the CICIDS2017 dataset, proving the quality over quantity principle for data-driven security products. The summarization module, powered by the T5-base transformer, also facilitated contextual understanding of every network flow, minimizing cognitive load to analysts by a considerable amount.

In summary, this project illustrates the growing promise of LLMs in cybersecurity—not just as generators of content, but also as intelligent readers of complex data. As these types of systems continue to be developed and validated, they may eventually become a regular component of next-generation Security Operations Centers (SOCs), enhancing analyst workflows and minimizing incident response times.

References

[1] S. Morgan, “Cybercrime to cost the world $10.5 trillion annually by 2025,” Cybersecurity Ventures, Nov. 2020. [Online]. Available: https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/

[2] National Institute of Standards and Technology (NIST), “Framework for Improving Critical Infrastructure Cybersecurity,” Version 1.1, Apr. 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

[3] MITRE Corporation, “MITRE ATT&CK Framework,” [Online]. Available: https://attack.mitre.org/

[4] MISP Threat Intelligence Platform. [Online]. Available: https://www.misp-project.org/

[5] S. V. Immanuel, M. Joseph, and R. Natarajan, “HuntGPT: Generative AI for Interpretable Cyber Threat Hunting,” *arXiv preprint*, arXiv:2309.16021, Sep. 2023.

[6] A. Basak and H. R. Shaikh, “Generative AI for Cyber Threat-Hunting in 6G-enabled IoT Networks,” arXiv preprint, arXiv:2303.11751, Mar. 2023.

[7] Canadian Institute for Cybersecurity (CIC), “CICIDS2017 Dataset,” University of New Brunswick, 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[8] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[9] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O’Reilly Media, 2009.

[10] T. Wolf et al., “Transformers: State-of-the-art Natural Language Processing,” in Proceedings of the 2020 EMNLP: System Demonstrations, pp. 38–45, 2020.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[12] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in Proc. 25th ICML, 2008, pp. 160–167.

[13] Scikit-learn Developers, “Feature selection using SelectKBest,” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[14] Hugging Face, “T5: Text-To-Text Transfer Transformer,” [Online]. Available: https://huggingface.co/t5-base

[15] OpenAI, “ChatGPT: Language Model for Dialogue,” [Online]. Available: https://openai.com/chatgpt

[16] AMD, “AMD Ryzen™ 7 5800H Mobile Processor,” [Online]. Available: https://www.amd.com/en/products/apu/amd-ryzen-7-5800h

[17] NVIDIA, “GeForce RTX 3060 Laptop GPU,” [Online]. Available: https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3060/

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. Int. Conf. Inf. Syst. Secur. Priv. (ICISSP), 2018. [Online]. Available: https://www.kaggle.com/d

[19] Lenovo, "Legion Pro 5 16IRX8 Specifications," 2024. [Online]. Available: https://psref.lenovo.com/syspool/Sys/PDF/Legion/Legion_Pro_5_16IRX8/Legion_Pro_5_16IRX8_Spec.pdf

[20] J. Desai, "Threat Hunting Using AI – Project Repository," GitHub, 2024. [Online]. Available: https://github.com/JhanviDesai/ThreatHuntingUsingAI

[21] M. Sushmitha, "Data Preprocessing Steps for Machine Learning in Python – Part 1," Medium, 2020. [Online]. Available: https://medium.com/womenintechnology/data-preprocessing-steps-for-machine-learning-in-phyton-part-1-18009c6f1153

[22] S. Punch, "Feature Importance Analysis in Machine Learning," Medium, 2022. [Online]. Available: https://medium.com/@SPUNCH/feature-importance-analysis-in-machine-learning-e0b5caf80ffc