International Journal of Advanced Scientific Research and Engineering Trends

ISSN: 2456-0774

Volume 10 Issue 6 (2026) | Pages: 1-8

A Lightweight AI-Driven Framework for Intelligent Cyber Threat Detection and Response

Gayatri Pandurang Bharne[1], Ankita Karale[2], Balkrishna K. Patil[3], Naresh Thoutam[4]

[1,2,3,4]Department of Computer Engineering, SITRC, Nashik-422213, India

[1]*gayatribharne26@gmail.com*, [2]*ankita.karale@sitrc.org*, [3]*balkrishnapatileng@gmail.com*, [4]*naresh.thoutam@sitrc.org*

Peer Review Information

***Type**: Article*
***Received**: 20 March 2026*
***Revised**: 12 April 2026*
***Accepted**: 22 May 2026*
***Published**: 10 June 2026*

Abstract

The rapid increase in cyberattacks has created a serious challenge for modern digital infrastructures. Traditional intrusion detection systems mainly depend on static rules and known attack signatures; therefore, they are often unable to detect zero-day attacks, polymorphic malware, advanced persistent threats, and newly emerging network anomalies. At the same time, cyber threat intelligence reports contain valuable Indicators of Compromise such as malicious IP addresses, suspicious domains, CVE identifiers, malware names, and attack patterns, but such information is usually available in unstructured textual form and remains underutilized in many academic intrusion detection systems. To address these limitations, this paper proposes ThreatSeer, a lightweight AI-driven framework for intelligent cyber threat detection and response. The proposed framework integrates structured intrusion detection system logs, unstructured cyber threat intelligence reports, automated IoC extraction, machine learning-based anomaly detection, lightweight knowledge graph representation, alert generation, visualization, and report export facilities. The system uses Python-based technologies such as SQLite, pandas, scikit-learn, XGBoost, Plotly, and NiceGUI to provide an offline-capable and academic-friendly solution. Random Forest and XGBoost models are used for detecting suspicious traffic patterns, while regex and NLP-assisted methods are used for extracting IoCs from threat reports. The extracted indicators and detection results are further converted into alerts and visualized through an interactive dashboard. Unlike enterprise-level security platforms that require large-scale cloud or big-data infrastructure, ThreatSeer is designed for standard laptops and small-scale deployment. The proposed framework contributes to cybersecurity research by combining detection accuracy, interpretability, visualization, and practical usability into one integrated system.

Keywords: Artificial Intelligence; Cybersecurity; Intrusion Detection System; Machine Learning; Threat Intelligence; Indicator of Compromise (IoC); Knowledge Graph; Random Forest; XGBoost; Natural Language Processing (NLP); SQLite; NiceGUI.

How to Cite This Article

Bharne, G. P., Karale, A., Patil, B. K., & Thoutam, N. (2026). A lightweight AI-driven framework for intelligent cyber threat detection and response. *International Journal of Advanced Scientific Research and Engineering Trends*, 10(6), 1–8.

Introduction

The twenty-first century has witnessed a major shift toward digital services, online platforms, connected devices, and data-driven operations. Organizations in banking, healthcare, education, e-commerce, government, and industrial sectors are increasingly dependent on digital infrastructure. This digital dependency has also increased exposure to cyber risks. Attackers now use advanced methods such as phishing, ransomware, malware injection, botnets, denial-of-service attacks, credential theft, data exfiltration, and advanced persistent threats. These attacks are not only increasing in number but also becoming more adaptive and difficult to detect.

Conventional cybersecurity systems such as firewalls, signature-based intrusion detection systems, and rule-based security information and event management tools are useful against known threats. However, their performance becomes limited when attacks are new, modified, obfuscated, or unknown. Signature-based systems mainly detect patterns that are already present in their database. If the attacker changes the attack structure or uses a zero-day vulnerability, such systems may fail to detect the threat in time. This creates a strong need for intelligent systems that can analyze network behavior, learn from data, detect abnormal patterns, and provide meaningful alerts.

Machine learning has emerged as an important approach for intrusion detection. Models such as Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, Gradient Boosting, XGBoost, and deep learning models have been widely used for classifying network traffic as normal or malicious [1]–[5]. Benchmark datasets such as KDD99, NSL-KDD, UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018 have supported the development and testing of intrusion detection methods [1]–[4]. However, many machine learning-based intrusion detection studies focus mainly on classification accuracy. They often ignore practical requirements such as threat intelligence integration, IoC extraction, alert explanation, visual analysis, report generation, and lightweight deployment.

Another important component in modern cybersecurity is cyber threat intelligence. Cyber threat intelligence reports provide useful information about attackers, malware campaigns, vulnerabilities, IP addresses, domains, hashes, CVEs, and tactics, techniques, and procedures. However, these reports are often written in natural language and cannot be directly used by a detection system. Therefore, automated extraction of Indicators of Compromise from unstructured reports is necessary. Natural language processing and regular expression-based extraction can convert unstructured CTI into structured data that can be stored, analyzed, and visualized.

To address these issues, this paper proposes ThreatSeer, a lightweight AI-driven framework for intelligent cyber threat detection and response. The framework combines structured IDS log analysis and unstructured CTI processing into one academic-friendly system. It uses machine learning models for anomaly detection, regex/NLP methods for IoC extraction, SQLite for local storage, and NiceGUI for dashboard-based interaction. A lightweight knowledge graph is also constructed to represent relationships between IoCs, alerts, and threat entities. The system is designed to run offline on student laptops and does not depend on cloud services, Hadoop, Spark, or enterprise-level infrastructure.

The major contributions of this paper are as follows:

- A lightweight integrated framework is proposed for AI-driven cyber threat detection and response.
- Structured IDS logs and unstructured CTI reports are combined in a unified workflow.
- Random Forest and XGBoost models are used for machine learning-based anomaly detection.
- IoCs such as IP addresses, domains, CVEs, and malware names are extracted using regex and NLP-assisted methods.
- A lightweight knowledge graph is designed to improve interpretability and threat relationship analysis.
- A NiceGUI-based dashboard is proposed for visualization, alert management, and report export.

Literature Review

Machine learning-based intrusion detection has been widely studied in cybersecurity research. Moustafa and Slay introduced the UNSW-NB15 dataset as a modern benchmark for evaluating network intrusion detection systems [1]. The dataset includes normal traffic and several attack categories such as DoS, Exploits, Generic, Reconnaissance, Backdoors, Shellcode, Worms, Analysis, and Fuzzers. This dataset is considered more realistic than older datasets because it includes modern attack behavior and flow-level network features.

Sharafaldin et al. proposed the CICIDS2017 dataset to provide realistic network traffic and diverse attack scenarios for intrusion detection evaluation [2]. The dataset includes attacks such as brute force, botnet, DoS, DDoS, infiltration, port scan, and web attacks. It has been widely used by researchers to evaluate supervised and deep learning-based IDS models. Similarly, CSE-CIC-IDS2018 extended this dataset family by providing larger and more diverse attack traffic for IDS research [3].

Earlier datasets such as KDD99 and NSL-KDD played an important role in the development of intrusion detection research. Tavallaee et al. introduced NSL-KDD to overcome some of the limitations of KDD99, especially duplicate records and biased evaluation [4]. However, these older datasets do not fully represent modern attack patterns. Therefore, recent studies prefer UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018 for evaluating current IDS models.

Breiman proposed Random Forest as an ensemble learning method that combines multiple decision trees to improve classification performance and reduce overfitting [5]. Random Forest has been frequently used in intrusion detection because it performs well on structured data and can handle nonlinear feature relationships. Chen and Guestrin introduced XGBoost, a scalable tree-boosting method that provides strong performance on tabular datasets [6]. Due to its efficiency and predictive power, XGBoost is also suitable for IDS classification problems.

Several survey studies have analyzed the role of machine learning in intrusion detection. Buczak and Guven reviewed data mining and machine learning methods for cybersecurity intrusion detection and highlighted the importance of feature selection, dataset quality, and evaluation metrics [7]. Garcia-Teodoro et al. reviewed anomaly-based network intrusion detection methods and explained that anomaly detection is useful for unknown attacks but may generate false positives [8]. Liao et al. provided a broad review of intrusion detection systems and discussed host-based, network-based, signature-based, and anomaly-based IDS approaches [9].

Apart from machine learning, cyber threat intelligence has become an important part of cybersecurity analysis. Threat intelligence helps security teams understand malicious entities, attack campaigns, vulnerabilities, and threat behavior. Standards such as STIX and TAXII support the structured exchange of cyber threat intelligence [10], [11]. MITRE ATT&CK provides a knowledge base of adversarial tactics and techniques, which helps analysts map observed behavior to known attack patterns [12]. These frameworks show the value of structured intelligence representation.

Knowledge graphs have also gained attention in cybersecurity because they can represent relationships between attackers, malware, vulnerabilities, IoCs, and incidents. Graph-based representation improves situational awareness and helps analysts understand how different threat entities are connected. Recent studies have explored knowledge graph construction from CTI reports and security data [13]–[15]. However, many such approaches require complex infrastructure or advanced graph databases. For academic and small-scale settings, a lightweight knowledge graph approach is more practical.

The reviewed literature shows that machine learning models can detect network anomalies effectively, and CTI frameworks can improve contextual understanding. However, many existing systems are fragmented. Some focus only on IDS classification, some focus only on CTI extraction, and some focus only on visualization. There is still a need for a single lightweight framework that combines IDS log processing, IoC extraction, ML detection, knowledge graph representation, alert management, and dashboard-based reporting. ThreatSeer is proposed to address this gap.

Research Gap and Problem Statement

Although many intrusion detection systems have been proposed, several research gaps remain.

First, traditional IDS solutions are often static and signature-based. These systems can detect known attacks but struggle with new attack variants, zero-day vulnerabilities, and polymorphic malware. Modern cyber threats are dynamic, and therefore detection systems must learn from data and adapt to new behavior.

Second, many machine learning-based IDS studies mainly report accuracy, precision, recall, and F1-score. Although these metrics are important, they are not sufficient for practical cybersecurity analysis. Security analysts also need information about why an alert was generated, which IoCs are involved, and how different threat entities are connected.

Third, cyber threat intelligence reports are underutilized in many academic IDS systems. These reports contain useful IoCs such as IP addresses, domains, CVE identifiers, and malware names. However, if this information remains in unstructured text format, it cannot directly support automated alert generation or threat visualization.

Fourth, many enterprise-level security platforms require large infrastructure, cloud services, distributed storage, or paid tools. Such systems are difficult to implement in academic laboratories or student projects. There is a need for a lightweight, offline-capable, and low-cost framework that can run on ordinary laptops.

Fifth, existing academic systems are often fragmented. One system may perform only ML classification, another may perform only CTI extraction, and another may focus only on visualization. A complete system must combine data ingestion, preprocessing, ML detection, IoC extraction, knowledge graph representation, alert management, visualization, and report generation.

Based on these gaps, the research problem of this study is defined as follows:

How can a lightweight AI-driven cybersecurity framework be designed to ingest heterogeneous cyber data, extract actionable threat intelligence, detect malicious network behavior using machine learning, represent threat relationships through a knowledge graph, and provide interpretable visualization and reporting for academic and small-scale deployment?

Objectives of the Study

The main objective of this study is to design and implement ThreatSeer, a lightweight AI-driven framework for intelligent cyber threat detection and response.

The specific objectives are:

- To develop a modular data ingestion pipeline for structured IDS logs and unstructured CTI reports.
- To normalize uploaded cyber data and store it in a unified SQLite database.
- To extract Indicators of Compromise such as IP addresses, domains, CVE identifiers, and malware names from CTI reports.
- To construct a lightweight knowledge graph for representing relationships between IoCs, alerts, and threat entities.
- To train machine learning models such as Random Forest and XGBoost for detecting malicious traffic patterns.
- To evaluate detection performance using accuracy, precision, recall, F1-score, and ROC-AUC.
- To generate alerts from both IoC-based matching and ML-based anomaly detection.
- To develop an interactive NiceGUI dashboard for visualization, alert management, and report export.
- To ensure that the system remains offline-capable, lightweight, and suitable for academic deployment.

Proposed ThreatSeer Framework

The proposed ThreatSeer framework follows a modular architecture. The system is divided into seven main modules: data ingestion, preprocessing and storage, IoC extraction, machine learning detection, knowledge graph construction, alert management, and visualization/reporting.

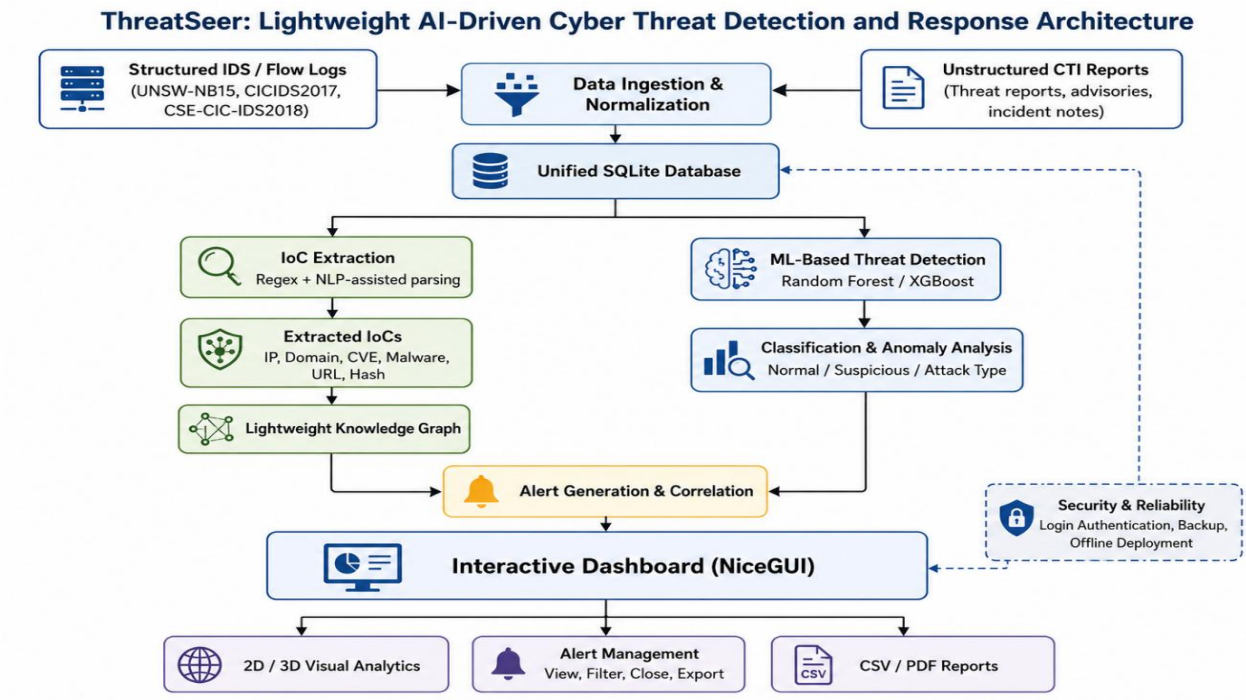

Fig. 1. *Proposed ThreatSeer architecture*

Data Ingestion Module

The data ingestion module accepts two types of input. The first input is structured IDS data in CSV or log format. This may include datasets such as UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018. These datasets contain network traffic records with features such as protocol, service, duration, packet count, byte count, source/destination details, and attack labels.

The second input is unstructured CTI text. This may include threat reports, malware descriptions, vulnerability notes, incident summaries, or manually prepared intelligence documents. These documents may contain IoCs and attack-related information that can support threat analysis.

Preprocessing and Storage Module

After ingestion, structured IDS data is cleaned and normalized. Missing values, duplicate records, categorical fields, and irrelevant attributes are handled. Categorical values may be encoded, and numerical features may be scaled depending on model requirements. The preprocessed data is stored in SQLite.

SQLite is selected because it is lightweight, serverless, offline-capable, and simple to integrate with Python applications. It is suitable for academic projects because it does not require database server configuration. The database stores uploaded files, cleaned records, extracted IoCs, model results, alerts, users, and reports.

IoC Extraction Module

The IoC extraction module processes unstructured CTI reports and extracts useful cybersecurity indicators. Regex patterns are used for detecting IP addresses, domains, CVE identifiers, URLs, file hashes, and email addresses. NLP-assisted preprocessing can be used for tokenization, text cleaning, keyword matching, and entity identification.

For example, a CTI report may contain a sentence such as: "The malware campaign exploited CVE-2021-44228 and communicated with the domain malicious-example.com." The IoC extraction module identifies the CVE identifier and suspicious domain, stores them in the database, and links them to a report ID.

Machine Learning Detection Module

The machine learning module is responsible for detecting malicious traffic from IDS datasets. Random Forest and XGBoost are selected as primary models because they are effective on structured tabular data and can run on CPU-based systems.

Random Forest works by combining multiple decision trees and generating a final prediction through majority voting. It is robust against overfitting and performs well on many classification tasks. XGBoost uses gradient boosting and improves prediction performance by sequentially correcting previous model errors.

The detection module classifies network traffic as normal or malicious. It may also support multi-class classification for attack categories such as DoS, Exploits, Reconnaissance, Backdoors, Botnet, Brute Force, or Web Attack depending on the dataset.

Knowledge Graph Module

The knowledge graph module represents relationships between extracted IoCs, alerts, traffic records, CVEs, domains, malware names, and attack types. The purpose of this module is to improve interpretability. Instead of only showing that an alert was generated, the graph can show which IoC triggered the alert, which CVE was involved, and which malicious domain or IP address was connected. A lightweight graph representation is preferred instead of a heavy graph database such as Neo4j. For academic demonstration, entities and relationships can be stored in SQLite and visualized using a graph library or Plotly-based network representation.

Alert Management Module

The alert management module combines IoC-based and ML-based detection results. An alert may be generated when a malicious IP or domain is detected in a CTI report, when an IDS record is classified as malicious by the ML model, or when an anomaly score crosses a defined threshold.

Each alert contains key metadata such as alert ID, timestamp, source, severity, detection reason, related IoC, model prediction, confidence score, and status. The dashboard allows users to view, filter, close, and export alerts.

Visualization and Reporting Module

The visualization module provides an interactive dashboard using NiceGUI. The dashboard includes key performance indicators, alert summaries, IoC statistics, model performance metrics, anomaly timelines, and graph-based threat relationships. 2D charts can show attack distribution, alert trends, and model performance. 3D visualizations can show traffic clusters or anomaly patterns.

The reporting module allows users to export results in CSV and PDF formats. Reports may include dataset summary, model metrics, extracted IoCs, generated alerts, and visualization snapshots. This feature is useful for academic documentation and project evaluation.

Methodology

The proposed methodology follows a systematic development and evaluation process.

Dataset Selection

The study considers benchmark intrusion detection datasets such as UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018. These datasets are selected because they include modern network attacks and are widely used in IDS research. The datasets contain labeled traffic records, making them suitable for supervised learning.

Data Preprocessing

The collected dataset is preprocessed before model training. Preprocessing includes handling missing values, removing duplicate entries, encoding categorical variables, normalizing numerical features, selecting relevant attributes, and splitting the dataset into training and testing sets. The processed data is stored in SQLite for reuse.

Model Training

Random Forest and XGBoost models are trained using the processed IDS dataset. Random Forest is used as a stable baseline model, while XGBoost is used for improved gradient-boosting performance. Hyperparameters such as number of trees, maximum depth, learning rate, and number of estimators can be tuned depending on dataset size and computational capacity.

Model Evaluation

The trained models are evaluated using standard performance metrics:

- Accuracy: Measures the overall percentage of correct predictions.
- Precision: Measures how many predicted malicious records are actually malicious.
- Recall: Measures how many actual malicious records are correctly detected.
- F1-score: Provides a balance between precision and recall.
- ROC-AUC: Measures the model's ability to distinguish between normal and malicious traffic.

These metrics help compare the performance of Random Forest and XGBoost.

CTI Processing and IoC Extraction

Unstructured CTI reports are processed using regex and NLP-based text cleaning. The system extracts IoCs such as IP addresses, domains, CVEs, URLs, malware names, and file hashes. Extracted indicators are stored in structured database tables and linked with reports or alerts.

Knowledge Graph Construction

Extracted IoCs and alert metadata are converted into graph nodes and edges. Nodes represent entities such as IP address, domain, CVE, malware, alert, or attack type. Edges represent relationships such as "related to," "triggered by," "reported in," or "associated with." This graph helps users understand the context of alerts.

Dashboard Development

A NiceGUI-based dashboard is developed to provide user interaction. The dashboard includes login, file upload, dataset view, model training, prediction results, alert table, IoC table, graph visualization, and export options. Plotly is used for 2D and 3D charts.

Expected Results and Discussion

The proposed ThreatSeer framework is expected to provide both technical and practical outcomes. From a detection perspective, Random Forest and XGBoost are expected to achieve strong baseline results on benchmark IDS datasets. The target performance is accuracy above 90% and F1-score above 88%, depending on dataset quality and preprocessing.

The IoC extraction module is expected to convert unstructured CTI reports into structured indicators. This will help bridge the gap between textual intelligence and automated detection workflows. Instead of manually reading reports and copying IoCs, the system can automatically identify and store useful indicators.

The knowledge graph is expected to improve interpretability. In many IDS systems, an alert only shows that traffic is malicious. However, ThreatSeer can show how an alert is related to a specific IP address, domain, CVE, or malware reference. This improves analyst understanding and makes the system more useful for cybersecurity education.

The dashboard is expected to improve usability. Students, researchers, or small organizations can use the dashboard to upload datasets, train models, extract IoCs, view alerts, and export reports. This makes the system more practical than a command-line-only ML project.

The lightweight design is also important. Many cybersecurity platforms depend on cloud infrastructure, distributed storage, or enterprise SIEM tools. ThreatSeer avoids such dependency and uses Python, SQLite, and NiceGUI. Therefore, it can run on ordinary laptops and can be demonstrated in academic laboratories.

Advantages of the Proposed System

The proposed system provides several advantages:

1. Integrated Workflow: It combines IDS log analysis, CTI processing, ML detection, IoC extraction, alerting, visualization, and reporting.
2. Lightweight Deployment: It does not require cloud services, Hadoop, Spark, or enterprise infrastructure.
3. Offline Capability: The system can work in local academic environments.
4. Improved Interpretability: Knowledge graph representation helps explain relationships between threats and indicators.
5. Academic Suitability: The framework is suitable for M.Tech, B.Tech, and research prototype demonstrations.
6. Extensibility: Additional datasets, ML models, IoC types, and visualization methods can be added in future.

7. Practical Dashboard**:** The NiceGUI interface makes the system user-friendly and easy to evaluate.

Limitations

Although the proposed framework is useful, it has some limitations. First, it is designed mainly for batch-mode analysis and does not focus on real-time enterprise-level packet monitoring. Second, the system depends on benchmark datasets, which may not fully represent live organizational traffic. Third, regex-based IoC extraction may miss complex or indirectly mentioned threat entities. Fourth, machine learning models may generate false positives if the dataset is imbalanced or if attack patterns are different from the training data. Fifth, the knowledge graph is lightweight and does not include advanced graph analytics.

Despite these limitations, the framework is suitable for academic implementation and small-scale cybersecurity analysis.

Conclusion

This paper proposed ThreatSeer, a lightweight AI-driven framework for intelligent cyber threat detection and response. The framework integrates machine learning-based IDS detection, cyber threat intelligence processing, automated IoC extraction, lightweight knowledge graph representation, alert management, visualization, and reporting. Random Forest and XGBoost are used for detecting malicious network traffic, while regex and NLP-assisted methods are used for extracting IoCs from CTI reports. SQLite provides lightweight local storage, and NiceGUI provides an interactive dashboard for user interaction.

The proposed system addresses several gaps in existing academic IDS projects. It moves beyond simple accuracy-based classification by adding threat intelligence, alert context, graph visualization, and report generation. The system is offline-capable, low-cost, and suitable for academic environments, student laptops, and small-scale organizational use.

Future work can extend the framework by adding Explainable AI methods such as SHAP and LIME, deep learning models such as CNN-LSTM, real-time log streaming, adversarial attack detection, and integration with standard CTI formats such as STIX and TAXII. Overall, ThreatSeer demonstrates that practical and interpretable cybersecurity tools can be developed using lightweight AI technologies.

References

1. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in *Proceedings of the Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
2. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “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, doi: 10.5220/0006639801080116.
3. Canadian Institute for Cybersecurity, “CSE-CIC-IDS2018 on AWS,” University of New Brunswick, 2018.
4. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.
5. L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
6. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
7. A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
8. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Computers & Security*, vol. 28, no. 1–2, pp. 18–28, 2009.
9. H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, “Intrusion detection system: A comprehensive review,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.
10. OASIS, “STIX Version 2.1: Structured Threat Information Expression,” OASIS Standard, 2021.
11. OASIS, “TAXII Version 2.1: Trusted Automated Exchange of Intelligence Information,” OASIS Standard, 2021.
12. MITRE Corporation, “MITRE ATT&CK: Adversarial Tactics, Techniques, and Common Knowledge,” MITRE, 2024.
13. S. Kim, J. Park, and D. Yoo, “LLM-TIKG: Large language model-driven threat intelligence knowledge graph construction,” *Computers & Security*, vol. 134, p. 103592, 2024.
14. J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, “TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data,” *Computers & Security*, vol. 95, p. 101867, 2020.
15. X. Liao, K. Yuan, X. F. Wang, Z. Li, L. Xing, and R. Beyah, “Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 755–766.

16. R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *Proceedings of the IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.
17. S. Axelsson, *Intrusion Detection Systems: A Survey and Taxonomy*, Technical Report, Chalmers University of Technology, 2000.
18. V. Jyothsna, V. V. Rama Prasad, and K. Munivara Prasad, “A review of anomaly based intrusion detection systems,” *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.
19. K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention Systems*, National Institute of Standards and Technology (NIST), Special Publication 800-94, 2007.
20. S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Boca Raton, FL, USA: CRC Press, 2011.