

Article

Automated Cyber Threat Intelligence Extraction from Distributed Honeypots: A Hybrid Machine Learning Approach

Hessa Abdulaziz AlJuhaيمان ¹, Qazi Emad-ul-Haq ^{1,2} , Kyounggon Kim ^{1,3,†}  and Seokhee Lee ^{1,3,*} 

¹ Department of Cybersecurity and Digital Forensics, College of Forensic & Investigative Sciences, Naif Arab University for Security Sciences (NAUSS), Riyadh 14812, Saudi Arabia; hessa.aljuhaيمان@hotmail.com (H.A.A.); qabdulrab@nauss.edu.sa (Q.E.-u.-H.); k.kim@rave.ac.uk (K.K.)

² Center of Artificial Intelligence for Security, Naif Arab University for Security Sciences (NAUSS), Riyadh 14812, Saudi Arabia

³ Center for Cybercrime and Economic Crime, Naif Arab University for Security Sciences (NAUSS), Riyadh 14812, Saudi Arabia

* Correspondence: slee@nauss.edu.sa

† Current address: Department of Computing and Emerging Technologies, Ravensbourne University London, London SE10 0EW, UK.

Abstract

The exponential growth of Indicators of Compromise (IoCs) has overwhelmed manual triage processes in Security Operations Centers (SOCs), necessitating automated solutions for large-scale log analysis. This study proposes a hybrid machine learning framework that integrates supervised and unsupervised learning to automate the classification, clustering, and contextual interpretation of Cyber Threat Intelligence (CTI). The primary contribution lies in a multi-stage feature engineering pipeline that enriches raw SIEM logs with cyclical temporal encoding and geographical metadata. In the supervised phase, a comparative evaluation of gradient boosting classifiers—XGBoost, LightGBM, and CatBoost—demonstrates that all three achieve competitive performance in categorizing known attack techniques, consistently outperforming the Random Forest baseline. The results indicate that classifier performance is dataset-dependent, and practitioners are encouraged to select the most suitable model based on their operational environment. Simultaneously, the unsupervised phase employs density-based clustering to identify emerging and previously unknown threat patterns by correlating adversarial behaviors with source attribution. By combining these two approaches, the framework ensures near-real-time feasibility and significantly enhances the scalability of automated threat extraction from distributed honeypot environments.

Keywords: Cyber Threat Intelligence (CTI); machine learning; hybrid framework; XGBoost; HDBSCAN; distributed honeypot; automated threat extraction



Academic Editor: Andreas Mauthe

Received: 19 April 2026

Revised: 9 June 2026

Accepted: 24 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid and continuous increase in cyber attacks highlights the importance of building a robust Cyber Threat Intelligence (CTI) platform to protect organizations from unexpected and evolving threats. CTI plays a key role in protecting organizations by providing valuable information about potential risks, attackers, and their capabilities [1]. By leveraging CTI, organizations can implement proactive defense strategies, shifting from a reactive posture to one that anticipates and neutralizes threats before they cause significant damage.

However, the exponential increase in cyber attacks is generating massive streams of threat data in the cybersecurity environment. Current CTI systems collect millions of Indicators of Compromise (IoCs) and log entries daily. Manually analyzing this vast amount of data is not only time-consuming but also prone to human error, making it difficult to ensure the speed required for effective incident response [2]. Since time is a sensitive and critical factor in mitigating cybercrime, the limitations of manual analysis—specifically the delay in processing and the inability to correlate vast datasets in real-time—pose a significant security risk.

To address these challenges, integrating machine learning (ML) into CTI platforms has become essential. ML technology can significantly reduce the time required for attack analysis by automating detection and classification processes. As Montasari et al. [3] highlight, utilizing CTI effectively supports the sharing of threat information in standardized, machine-readable formats such as STIX (Structured Threat Information eXpression), which can be technically integrated with security platforms to enhance defenses against probable threats.

Recent advancements have further demonstrated the value of AI-based approaches in this field. Guo et al. [4] proposed an adversarial attack detection method based on bidirectional consistency discrimination for deep learning-based soft sensors, while Spyros et al. [5] introduced an AI-based holistic framework for comprehensive CTI management. These works underscore the growing need for automated and scalable threat intelligence solutions.

1.1. Background and Motivation

In the current situation, it is necessary to rapidly process the vast amount of data generated by honeypots and Security Information and Event Management (SIEM) systems to identify the severity and characteristics of threats. While CTI contains extensive information about threats, extracting practical insights requires significant effort from security analysts. Machine learning is utilized because it can shorten this analysis time, predict threats, and accurately identify the most frequent sources of attacks targeting the organization.

Existing CTI research has primarily focused on classifying malware types or identifying known attack signatures using supervised learning methods, paying limited attention to identifying behavioral attribution and clustering geographical sources. Consequently, there is a lack of a comprehensive perspective on the sources of cyber attacks within the CTI framework. Responding rapidly to attacks and identifying their geographical and infrastructural sources is critical for enterprises to prevent customer information from leaking into the black market and to avoid service disruptions. Therefore, utilizing machine learning not merely for detection but also to identify attack sources and cluster similar attack patterns is essential for an integrated defense strategy.

1.2. Problem Statement

The core problem addressed in this study is the high-volume alert stream challenge inherent in modern CTI ecosystems. As organizations integrate diverse log sources into SIEM platforms, the volume of security events has surpassed the cognitive capacity of human analysts. Brown and Lee [2] highlighted that the sheer velocity and volume of threat data often lead to “alert fatigue,” causing critical indicators to be overlooked amidst false positives.

Despite the advancements in Machine Learning, significant gaps remain in current methodologies:

- **Limitations of Supervised Learning:** Most existing studies focus heavily on supervised classification for Intrusion Detection Systems (IDS), identifying known attack signa-

tures [6]. However, these models struggle to detect novel, evolving threat campaigns (Zero-day attacks) that lack labeled historical data.

- **Lack of Behavioral Attribution:** Researchers have placed less emphasis on predicting the source infrastructure and grouping attacks based on multidimensional behaviors. Traditional analysis often treats alerts in isolation rather than correlating them to identify broader campaigns, such as region-specific botnets or coordinated credential stuffing [7].
- **Latency in Incident Response:** Manual triage processes are prohibitively time-consuming. Montasari et al. [3] emphasized that without automated mechanisms to cluster and prioritize threats, the window of exposure to cyber risks significantly increases.

To bridge these gaps, this study proposes a hybrid framework that not only classifies known threats but also autonomously groups unknown patterns. The central research question this paper aims to answer is: “How can supervised and unsupervised Machine Learning be synergistically utilized to classify attack origins and cluster hidden threat patterns using IoCs, thereby enhancing cybercrime analytics and mitigating risk?”

1.3. Contributions

This study demonstrates the efficacy of a hybrid machine learning framework that integrates supervised and unsupervised learning to enhance automated threat intelligence. The main contributions of this paper are as follows:

- **Real-World Distributed Evaluation Dataset:** Unlike studies relying on synthetic or outdated traffic, this study validates the proposed framework using a dataset collected directly from a globally distributed honeypot infrastructure, capturing authentic adversarial behaviors and diverse attack vectors from live threat landscapes. The dataset is available on request from the corresponding author, subject to applicable data sharing agreements and institutional data governance policies.
- **Hybrid Learning Framework for Known and Unknown Threats:** We propose a dual-stage architecture that combines supervised classification for known attack techniques with unsupervised clustering for anomaly discovery. This hybrid approach ensures both high-precision categorization of established threats and the proactive identification of emerging, previously unseen attack patterns.
- **Optimized Feature Engineering for CTI:** We developed a specialized feature engineering pipeline that incorporates cyclical temporal encoding and geographical metadata. Our results demonstrate that gradient boosting classifiers, leveraging these enriched features, achieve competitive stability in addressing the severe class imbalances inherent in real-world CTI data. Among the evaluated models, CatBoost achieves the highest Balanced Accuracy of 0.7895 on Dataset 1, while XGBoost and LightGBM demonstrate superior F1-Macro performance across other configurations. Classifier selection should be guided by the specific dataset and operational requirements of the target SOC environment.
- **Behavioral Pattern and Campaign Discovery:** By implementing density-based clustering, we successfully identified coordinated threat campaigns and hidden behavioral groups. This module correlates agent behaviors with source attribution, enabling the discovery of region-specific botnets and automated attack infrastructures that bypass traditional rule-based detection.
- **Operational Efficiency in SOC Environments:** We validated the framework’s suitability for real-time triage, achieving sub-minute inference latency across large-scale datasets. This ensures that the proposed system can be integrated into Security Op-

erations Centers (SOCs) to reduce analyst workload and improve the mean time to detect (MTTD).

The remainder of this paper is organized as follows. Section 2 reviews related work in CTI and machine learning. Section 3 describes the datasets used. Section 4 details the proposed methodology. Section 5 presents the experimental results, followed by a discussion in Section 6. Finally, Section 7 concludes the paper with recommendations for future research.

2. Literature Review

In the cyber security field, CTI has emerged as a vital tool for establishing a unified platform against threats. With the exponential increase in published threat information, the necessity for automated analysis has become paramount. Machine learning offers a robust approach to address this challenge. This section reviews existing research in two main areas: Cyber Threat Intelligence and Machine Learning applications in cybersecurity.

2.1. CTI

Creating an operative platform to prevent cybercrime is a critical objective for modern organizations. By understanding the sources and mechanisms of threats, organizations can activate and implement defensive systems, thereby protecting themselves from potential cybercrime.

2.1.1. CTI Platforms and Sharing

To better detect and prevent cybercrime threats, organizations make concerted efforts to define practices to protect against complex attacks. This collaboration involves sharing threat-related information, such as IoCs. IoCs are specific forensic data elements that provide detailed information about illegal and malicious activities on a system or network [3]. According to Wagner et al. [1], threat intelligence meetings occur in specific locations to facilitate this sharing. For instance, NC4 CTX/Soltra Edge focuses on sharing financial CTI, while platforms such as the Malware Information Sharing Platform (MISP), ThreatConnect, AlienVault, and ThreatQuotient focus primarily on sharing General CTI. Recent studies by Alzahrani, Lee, and Kim have further demonstrated the efficacy of leveraging IoCs and MISP integration to enhance CTI capabilities, specifically within the Arab world context [8].

2.1.2. Data Sources for CTI

While CTI can be derived from various types of intelligence disciplines such as Human Intelligence (HUMINT) and Open-Source Intelligence (OSINT), technical security data remains the most critical source for automated analysis. In the context of cybersecurity, Cyber-Intelligence (CYBINT) focuses on data gathered from cyber domains, including SIEM logs, intrusion detection alerts, and honeypot telemetry [9]. Unlike subjective human intelligence, these technical sources provide structured IoCs, such as IP addresses, file hashes, and URL signatures which are essential for training machine learning models to detect anomaly patterns [3]. Consequently, this study focuses on extracting high-fidelity intelligence directly from raw SIEM logs generated by distributed honeypots.

2.1.3. CTI Benefits

Effectively implemented, CTI offers significant advantages by enabling the sharing of threat information in machine-readable formats. It supports organizations in developing targeted defenses and provides valuable insights for applying appropriate cybersecurity tools and solutions for protection [10]. In a survey conducted by Brown and Lee [2], respondents were asked about the effectiveness of CTI in improving their security and response.

The results revealed that 81% of the respondents responded positively, acknowledging that CTI had indeed enhanced their security measures. On the other hand, 17% of respondents were unsure about the impact of CTI, while only a mere 2% found CTI to be unhelpful.

2.1.4. Honey pots

Honey pots are critical systems for enhancing cybercrime research and analysis. They enable analysts to set up environments to trap attackers and discover the newest patterns and techniques used to exploit vulnerabilities [11]. Honey pots can run on different operating systems such as Windows and Linux, and support the collection of attack data for sharing on CTI platforms. The advantages organizations can gain by deploying a honey pot include gathering attack information, identifying malicious actors, capturing attack patterns, and providing new threat intelligence [12].

Recent research has significantly advanced honey pot methodologies. Franco et al. [13] conducted a comprehensive survey of honey pots and honeynets across IoT, Industrial IoT, and cyber-physical systems, demonstrating their critical role in modern threat intelligence gathering. Vetterl and Clayton [14] proposed a virtual honey pot framework for capturing CPE and IoT attacks, highlighting the evolving role of honey pots in modern threat landscapes. By analyzing the IoCs collected from these environments, cyber analysts can share actionable intelligence on CTI platforms and implement targeted defensive actions. More recently, El Kouari et al. [15] proposed a robust IIoT cybersecurity architecture integrating vertical honey pots across all Industry 4.0 levels with Wazuh for log transmission and CTI integration, further demonstrating the expanding role of honey pots in modern industrial threat intelligence environments.

2.2. Machine Learning in Cybersecurity

ML, a subfield of Artificial Intelligence, uses statistical techniques to develop algorithms that learn directly from existing datasets rather than relying on explicitly written code. ML has been extensively combined with cybersecurity expertise to improve strategies against CTI.

2.2.1. Supervised Learning Approaches

A wide range of advancements in cybersecurity-related ML has been covered in the literature. Alqahtani et al. examined the integration of ML techniques for developing robust Intrusion Detection Systems (IDS), testing Bayesian Networks, Decision Trees (DT), and Artificial Neural Networks (ANN) across multiple cybersecurity datasets [6]. Similarly, Al-Mhiqani et al. provided a complete analysis of insider threat detection, highlighting taxonomy classification, ML methods, datasets, and challenges [16]. Abu Al-Haija introduced a top-down ML-based architecture for the detection and classification of cyberattacks in IoT communication networks, emphasizing state-of-the-art techniques [17]. Asif et al. introduced a new approach named MapReduce-Based Intelligent Model for Intrusion Detection (MR-IMID), achieving an accuracy of 95.7% [18].

Specific to threat attribution and classification:

- Phishing: Alam et al. emphasized the value of ML in thwarting social engineering threats, achieving the best accuracy of 97% using Random Forest for phishing attack detection [19].
- IoT Security: Al-Hawawreh et al. offered a deep learning-based threat intelligence algorithm (DLTI) designed for complex IoT networks, comparing it with K-Nearest Neighbors (KNN), Naïve Bayes (NB), and Logistic Regression (LR) [20]. Mishra et al. used several models for IoT CTI, reporting high accuracies of 99.94% and 95.67% for Random Forest and KNN, respectively [21].

- Framework Vulnerabilities: Khurana et al. investigated poisoning attacks on AI-based threat intelligence systems. They used an ensembled semi-supervised approach combining an embedding model and an SVM model, achieving 71.73% accuracy [22].
- Data Gathering: Koloveas et al. advanced threat intelligence gathering by introducing “intime,” an ML-based framework for obtaining and utilizing web data for CTI [23].
- Deep Learning: Lee et al. emphasized the use of artificial neural networks for cyber threat detection, highlighting ML’s significance in identifying intricate attack patterns [24].
- Attribution: Noel introduced “RedAI,” utilizing Naïve Bayes, Logistic Regression, Linear SVM, and Random Forest. The highest accuracy achieved was 93.65% using Linear SVM [25]. Noor et al. achieved 95% accuracy in attributing cybercrime threats using ANNs, demonstrating the effectiveness of high-level IoCs over low-level IoCs [26].
- Mobile Security: Tahtaci and Canbay used Random Forest and Decision Trees to detect malware on Android, achieving an accuracy of 95% with Random Forest [27].

2.2.2. Unsupervised Learning Approaches

Unsupervised machine learning is used to discover relationships or patterns in a dataset without clear supervision or labeled examples. The K-means clustering algorithm is widely favored for its ability to handle complex datasets and data with high similarity. K-means has been applied to network intrusion detection by grouping traffic samples according to distance to cluster centers, enabling efficient identification of attack categories from log data [28]. Since CTI involves large datasets, the processing efficiency of K-means is a significant advantage [29].

In log analysis, Riadi et al. proposed K-means to cluster network attacks into three categories from log files, achieving sufficient results in grouping attacks based on criteria like used ports and TCP flags [28]. Sinaga and Yang tested K-means with different datasets, reporting good performance and high accuracy [30]. Zaid Mustafa and Rashid Amin applied various unsupervised ML methods, finding ML techniques to be effective, as demonstrated in a comprehensive survey [31]. Yoga et al. used the UNSW-NB15 dataset with K-means, proposing a unified framework to prevent cybercrime threats [32].

2.3. Summary and Implications

Based on the literature review, this study addresses the limitations of traditional single-model approaches by implementing a hybrid machine learning framework. We have selected XGBoost as our primary supervised learning algorithm due to its superior performance in handling the extreme class imbalances and high-dimensional feature sets inherent in real-world CTI data [33]. Previous studies have utilized various classifiers such as KNN, Random Forest, SVM, and ANN, but there is still a lack of systematic evaluation of gradient boosting methods for CTI log classification under conditions of severe class imbalance.

For unsupervised learning, we transition from traditional centroid-based methods like K-means to Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). This choice is motivated by the need to identify clusters of varying densities and effectively filter out noise, which is critical when analyzing unpredictable adversarial behaviors. Although prior research has utilized machine learning for intrusion detection, a significant gap remains in the contextual integration of temporal and geographical features to visualize threat origins. This study aims to bridge this gap by combining these optimized supervised and unsupervised techniques, thereby enhancing actionable cybercrime analytics and strategic risk mitigation. In contrast to existing hybrid CTI frameworks that typically combine supervised classifiers with static clustering algorithms on benchmark

datasets, our approach introduces: (i) cyclical temporal encoding with geo-contextual enrichment, (ii) HDBSCAN for noise-aware density-based campaign discovery, and (iii) live-capture distributed honeypot telemetry providing authentic adversarial diversity unavailable in synthetic datasets.

3. Data Description

All data utilized in this study were collected directly from a distributed honeypot infrastructure operated by our research group. To enhance the research's originality and focus on contemporary cyber threats, this study exclusively utilizes a real-time threat stream captured from a multi-regional testbed deployed in 2025. By utilizing this live-capture dataset, rather than relying on synthetic or pre-processed benchmark datasets, we aim to validate the effectiveness of the proposed hybrid framework—integrating gradient boosting classifiers for high-performance supervised classification and HDBSCAN for density-based unsupervised clustering—within a modern Security Operations Center (SOC) context.

3.1. Feature Engineering and Preprocessing

To ensure the model's robustness against time-zone variations and temporal cycles, we applied rigorous preprocessing steps:

- **UTC Standardization:** All timestamps from the distributed agents (US and Saudi Arabia) were normalized to Coordinated Universal Time (UTC) to align event sequences across different time zones.
- **Cyclical Time Encoding:** Simply treating 'Hour' as a linear integer (0–23) creates a logical discontinuity (e.g., 23:00 is far from 00:00 numerically but close temporally). To resolve this, we transformed time features into cyclical coordinates using Sine and Cosine functions:

$$x_{sin} = \sin\left(\frac{2\pi \times hour}{24}\right), \quad x_{cos} = \cos\left(\frac{2\pi \times hour}{24}\right) \quad (1)$$

This ensures that the model correctly interprets the continuity of time.

- **Categorical Encoding:** To support high-dimensional feature spaces required for gradient boosting classifiers and HDBSCAN, categorical variables such as agent.name and rule.description were transformed using One-Hot Encoding. This prevents the models from inferring incorrect ordinal relationships while preserving the distinct identity of each feature.

3.2. Data Collection Environment

To capture authentic cyber threats, we established a distributed monitoring infrastructure using the Wazuh SIEM platform. The monitoring infrastructure consisted of four distinct agents deployed in two geographically separated regions to collect various types of threat intelligence, utilizing a distributed honeypot architecture consistent with established honeypot deployment methodologies [11,12].

- **United States Zone (US Zone):** This region hosted two agents, wazuhOT and deepfake_v2_us. These agents were deployed on public network segments (IP range 199.168.x.x), exposing them directly to the open internet. Consequently, this zone attracted the highest volume of automated scanning and brute-force attacks.
- **Saudi Arabia Zone (Riyadh):** This region included agents deepfake_v1 and koko. Unlike the US zone, these agents were deployed within a private network environment (IP range 192.168.x.x) in Riyadh. Despite sitting behind a firewall/NAT, these nodes exhibited a unique attack profile, specifically targeted by persistent threat actors originating from specific regions such as Russia.

3.3. Distributed Threat Stream Overview

The dataset comprises security alerts processed by the Wazuh manager over a period of six weeks, from 24 May 2025 to 2 July 2025. To ensure analytical stability, the collection captured three distinct bursts of malicious activity, each containing exactly 10,000 events. The final dataset consists of a total of 30,000 unique security events. The temporal distribution is characterized by three specific high-intensity attack windows:

- Phase 1: 24–26 May 2025 (10,000 events).
- Phase 2: 1–2 June 2025 (10,000 events).
- Phase 3: 1–2 July 2025 (10,000 events).

3.4. Operational Triage Gap and Research Motivation

An initial statistical analysis of the threat stream reveals a landscape dominated by authentication-based attacks, with over 95% of events targeting Secure Shell (SSH) services. While these statistics illustrate the volume of activity, relying solely on descriptive metrics presents significant challenges for SOC operations.

The staggering magnitude of repetitive, low-severity alerts contributes to “alert fatigue,” making it arduous for analysts to distinguish between automated background noise and sophisticated, targeted campaigns [34]. Raw frequency counts do not provide actionable insights into the intent or anomaly of specific behaviors, leaving security teams without a clear basis for prioritization.

To overcome this bottleneck, our proposed framework utilizes gradient boosting classifiers for rapid, high-accuracy classification of known techniques and source countries, while employing HDBSCAN to automatically identify anomalous patterns and coordinate clusters within this noise. This hybrid approach transforms raw telemetry into actionable intelligence, optimizing decision-making in real-time environments.

4. Methodology

This study adopts a comprehensive, dual-track machine learning pipeline to analyze real-time CTI data effectively. The methodology follows a structured process involving data pre-processing, hybrid feature engineering, and the integrated application of state-of-the-art supervised and unsupervised learning models to classify known attack techniques and discover latent threat patterns.

All experiments were conducted using Python (version 3.11.7), lightgbm (version 4.6.0), catboost (version 1.2.10) and the XGBoost library (version 3.2.0) on a system equipped with an Intel Core i7 CPU (Intel Corporation, Santa Clara, CA, USA) and 16 GB RAM, running Windows 11 (64-bit). The data was collected using the Wazuh SIEM platform, and the overall architecture of the proposed pipeline is illustrated in Figure 1.

4.1. Phase 1: Data Pre-Processing and Sampling

To ensure the quality and operational relevance of the real-time threat stream, we performed rigorous cleaning and selection steps. The initial raw telemetry was filtered to remove rows and columns with predominantly null values or single-value dominance (e.g., program_name) to eliminate noise [35].

Based on domain expertise and statistical correlation, the dimensionality was reduced to 8 key features: source IP (data.srcip), target agent (agent.name), timestamp, fired rule count, destination user (data.dstuser), country (GeoLocation.country_name), rule description, and the MITRE ATT&CK technique (rule.mitre.technique) [36]. To address the inherent class imbalance where specific authentication failures dominated the stream, we applied a random under-sampling strategy to achieve a balanced distribution across primary attack categories, ensuring that the subsequent supervised models remain unbiased.

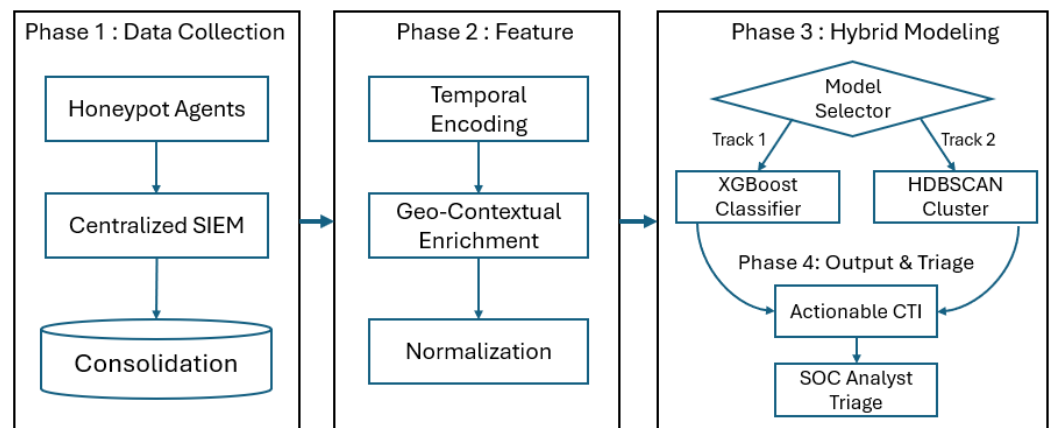


Figure 1. Proposed Machine Learning Pipeline for CTI Analysis, illustrating the process of supervised classification and unsupervised clustering.

4.2. Phase 2: Hybrid Feature Engineering and Encoding

To optimize the feature space, we implemented a specialized PipelineTransformer that captures environmental and temporal circumstances through the following strategies:

- **Cyclical Temporal Transformation:** To preserve the continuous nature of time, the event hour is mapped to sine and cosine coordinates ($hour_{sin}$, $hour_{cos}$). This mathematical transformation ensures that 23:00 and 00:00 are treated as temporally adjacent.
- **Numerical Identity Mapping:** Categorical metadata, including destination username and agent name, are transformed into numerical vectors using label encoding or hashing. This allows the model to process non-numeric identity context.
- **Network Infrastructure Representation:** Source IP addresses are converted to their integer equivalents ($data.srcip_{int}$) to preserve implicit subnet proximity in the feature space. Future work will explore AS-level embeddings as a richer network infrastructure representation.
- **Behavioral and Geo-Spatial Contextualization:** We quantified the recurrence of alerts via rule.firedtimes and derived geographical origin metadata ($GeoLocation.country_name_encoded$) from IP-to-location mapping.

4.3. Experimental Feature Configurations

To evaluate the impact of different contextual layers on model performance, we defined three incremental feature sets. This allows us to observe how the inclusion of situational and geographical metadata contributes to the accuracy of threat detection. The specific composition of each set is summarized in Table 1.

- **Minimal (IP + Time):** Represents the baseline telemetry focusing on the fundamental “when” and “where” (at the network level). It utilizes only the integer-converted source IP and cyclical temporal coordinates.
- **Contextual:** Enhances the baseline by adding the “who” and “how” of an event. This set includes target user identity, reporting agent profiles, and rule firing frequency to capture the behavioral footprint of the adversary.
- **Contextual + Geo:** The most comprehensive configuration, incorporating geographical intelligence. By adding encoded country-level data, the model can correlate behavioral patterns with regional anomalies.

Table 1. Summary of Feature Components across Experimental Sets (✓ indicates the inclusion of the corresponding feature).

Feature Name	Minimal	Contextual	Contextual + Geo
Source IP (<i>srcip_int</i>)	✓	✓	✓
Temporal (<i>sin/cos</i>)	✓	✓	✓
Destination User		✓	✓
Agent Name		✓	✓
Rule Fired Times		✓	✓
Geo Location (Country)			✓

4.4. Phase 3: Hybrid Modeling Framework (Track 1 & Track 2)

We propose a synergistic dual-track approach that combines the precision of supervised learning with the discovery capabilities of unsupervised learning.

4.4.1. Track 1: High-Performance Classification

For the rapid classification of known attack techniques and source countries, we evaluate four supervised classifiers: XGBoost, LightGBM, CatBoost, and Random Forest. All four models were evaluated under identical experimental conditions. XGBoost, LightGBM, and CatBoost were optimized using GridSearchCV with 5-fold cross-validation, tuning hyperparameters such as *learning_rate*, *n_estimators*, and *max_depth*. The hyperparameter configurations for all models are summarized in Table 2.

An initial performance comparison between XGBoost and the Random Forest baseline was conducted, as detailed in Table 3. Furthermore, as shown in the extended results in Table 4, all three gradient boosting models consistently outperform Random Forest across all datasets and feature configurations, while exhibiting dataset-dependent performance differences among themselves.

Random Forest represents a widely used bagging-based ensemble baseline in cybersecurity, while XGBoost, LightGBM, and CatBoost are state-of-the-art gradient boosting frameworks. However, no single gradient boosting model dominates across all configurations, indicating that classifier selection should be tailored to the specific dataset and SOC environment. XGBoost was used as the primary classifier in this study's pipeline; however, the framework is designed to accommodate any of the evaluated gradient boosting models.

Table 2. Hyperparameter Configurations for All Supervised Models and HDBSCAN.

Model	Parameter	Value
XGBoost	<i>n_estimators</i>	500
	<i>max_depth</i>	6
	<i>learning_rate</i>	0.05
	<i>subsample</i>	0.9
	<i>colsample_bytree</i>	0.9
	<i>reg_lambda</i>	1.0
	<i>objective</i>	multi:softprob/binary:logistic
LightGBM	<i>n_estimators</i>	500
	<i>max_depth</i>	6
	<i>learning_rate</i>	0.05
	<i>subsample</i>	0.9
	<i>colsample_bytree</i>	0.9
	<i>class_weight</i>	Inverse-frequency weighted (<i>class_w</i>)
	<i>objective</i>	multiclass/binary

Table 2. *Cont.*

Model	Parameter	Value
CatBoost	iterations	500
	depth	6
	learning_rate	0.05
	bootstrap_type	Bernoulli
	subsample	0.9
	auto_class_weights	Balanced
	objective	MultiClass/Logloss
HDBSCAN	min_cluster_size	10
	min_samples	5
	metric	euclidean
	cluster_selection_method	eom

Table 2 summarizes the hyperparameter configurations used for all models in all experiments.

Table 3. Performance Comparison of XGBoost and Random Forest across Multiple datasets.

Dataset	Model	Accuracy	Balanced Acc.	Recall	F1-Score
Dataset 1 (Geo)	Random Forest	0.5560	0.6348	0.6242	0.5534
	XGBoost	0.6015	0.7712	0.6849	0.6005
Dataset 2 (Geo)	Random Forest	0.7269	0.6712	0.6357	0.7248
	XGBoost	0.7289	0.7696	0.6447	0.7354
Dataset 3 (Context)	Random Forest	0.5555	0.5952	0.5997	0.5558
	XGBoost	0.5860	0.7261	0.6501	0.5863

Table 4. Extended Performance Comparison of All Models.

Dataset	Model	Accuracy	Balanced Acc.	F1-Macro	F1-Weighted
Dataset 1 (Geo)	Random Forest	0.5560	0.6348	0.6242	0.5534
	XGBoost	0.5975	0.7628	0.6806	0.5965
	LightGBM	0.5995	0.7593	0.6756	0.5994
	CatBoost	0.6000	0.7895	0.6571	0.5931
Dataset 2 (Geo)	Random Forest	0.7271	0.6478	0.6563	0.7242
	XGBoost	0.7326	0.7653	0.7183	0.7384
	LightGBM	0.7284	0.7221	0.6653	0.7339
	CatBoost	0.6893	0.8178	0.6044	0.7003
Dataset 3 (Context)	Random Forest	0.5540	0.5897	0.5898	0.5538
	XGBoost	0.5770	0.7212	0.6506	0.5769
	LightGBM	0.5885	0.7255	0.6522	0.5891
	CatBoost	0.5815	0.7693	0.6470	0.5788

95% Bootstrap Confidence Intervals (1000 resamples, `random_state = 42`). Dataset 1: RF BAcc [0.5935, 0.6743], XGBoost BAcc [0.7348, 0.7883], LightGBM BAcc [0.7262, 0.7878], CatBoost BAcc [0.7719, 0.8057]. Dataset 2: RF BAcc [0.6070, 0.6992], XGBoost BAcc [0.7085, 0.8267], LightGBM BAcc [0.5959, 0.8005], CatBoost BAcc [0.7147, 0.8858]. Dataset 3: RF BAcc [0.5360, 0.6347], XGBoost BAcc [0.6710, 0.7615], LightGBM BAcc [0.6770, 0.7663], CatBoost BAcc [0.7330, 0.7907]. Full CI values for all metrics are available upon request.

4.4.2. Track 2: Density-Based Pattern Discovery

To identify previously unknown threat campaigns within the noise, we transitioned from traditional K-means to HDBSCAN. Unlike K-means, which requires a pre-specified number of clusters and assumes spherical distributions, HDBSCAN can discover clusters of varying shapes and sizes while explicitly identifying low-density noise points [37]. This

is particularly effective for isolating rare or emerging adversarial behaviors that do not conform to established signatures.

4.5. Validation Framework

To ensure statistical reliability, the framework was validated using a 20% hold-out test set that was completely isolated from the training and hyperparameter tuning phases. We utilized 5-fold cross-validation during the training of the gradient boosting classifiers to prevent overfitting. Performance was evaluated not only based on accuracy and F1-scores but also on Inference Latency, ensuring that the hybrid system can provide actionable intelligence within the time constraints of an operational Security Operations Center.

4.5.1. Algorithm Selection Rationale

The selection of the gradient boosting classifier and HDBSCAN reflects a shift toward scalability and interpretability. the gradient boosting classifier provides a robust baseline for known threats with high computational efficiency. Concurrently, the integration of HDBSCAN addresses the “Zero-day” gap, grouping anomalous events that deviate from the patterns identified in Track 1. This hybrid architecture ensures maximum coverage of the threat landscape while significantly reducing the cognitive load on SOC analysts by suppressing repetitive background noise.

4.5.2. Decision-Level Integration

In the experimental phase of this study, Track 1 (gradient boosting classifier) and Track 2 (HDBSCAN) operate independently and in parallel. This parallel evaluation was designed to rigorously assess the standalone capabilities of each algorithm, ensuring that HDBSCAN’s ability to discover hidden threat campaigns was not biased by upstream classification filtering. Together, they provide a comprehensive threat response: Track 1 ensures rapid identification of existing threats, while HDBSCAN uncovers zero-day behavioral patterns that deviate from known signatures.

However, for practical deployment in a high-volume Security Operations Center (SOC) environment, a cascaded (sequential) decision-level integration is highly recommended. In an operational pipeline, samples confidently classified as known attacks by the gradient boosting model (e.g., routine authentication failures) would be automatically filtered and routed for immediate mitigation. HDBSCAN would then exclusively process the residual—unclassified or low-confidence samples. This cascaded approach eliminates redundant processing of known threats, significantly reducing computational overhead while maximizing the discovery of novel campaigns.

5. Experimental Results

5.1. Supervised Learning Results: Classification and Attribution

In this section, we evaluate the performance of the proposed gradient boosting classifiers against a Random Forest baseline. The experiments validate the model’s effectiveness in a high-velocity CTI environment, focusing on its ability to handle class imbalance while maintaining high classification accuracy.

5.1.1. Experimental Setup and Iterations

To ensure the statistical reliability and generalizability of the results, a total of 54 systematic experiments were conducted under identical data splits for the primary XGBoost and Random Forest evaluation. Each dataset (Dataset 1, 2, and 3) underwent 18 distinct experimental runs, which were symmetrically divided into 9 combinations for XGBoost and 9 for Random Forest across different feature configurations. In addition, an extended evalu-

ation was conducted to include LightGBM and CatBoost as additional baseline classifiers. To ensure robust statistical validation, the final performance metrics were derived from a single stratified 80/20 test split using 95% Bootstrap Confidence Intervals (1000 resamples).

A key aspect of this evaluation was the use of multiple classification targets to assess the models' adaptability across different levels of the threat hierarchy. The experiments were performed against three distinct target variables:

- Rule Description: To evaluate the model's ability to classify raw SIEM alert semantics.
- MITRE ATT&CK Tactic: To assess high-level adversarial goals.
- MITRE ATT&CK Technique: To validate granular-level classification of specific attack methods.

Combined with 5-fold cross-validation during each run, these 54 iterations across diverse targets ensure that the performance metrics reported in Table 3 represent a comprehensive and stable evaluation of the hybrid framework's capabilities, rather than a narrow optimization for a single classification task.

The visual evidence supporting this stability is presented. Specifically, Figure 2 demonstrates the performance advantage of XGBoost over Random Forest in dataset 1, particularly in achieving a Balanced Accuracy of 0.7712. The Classification fidelity of the model is further validated by the confusion matrix in Figure 3, which shows clear diagonal dominance across multi-class attack techniques in dataset 2. Feature importance was extracted using XGBoost's scikit-learn compatible `feature_importances_` attribute, which derives each feature's importance score based on its contribution across all decision tree splits. As shown in Figure 4, the high classification accuracy is driven by the strategic integration of geographical and temporal features, with variables such as `hour_cos` and `GeoLocation` appearing as top predictors.

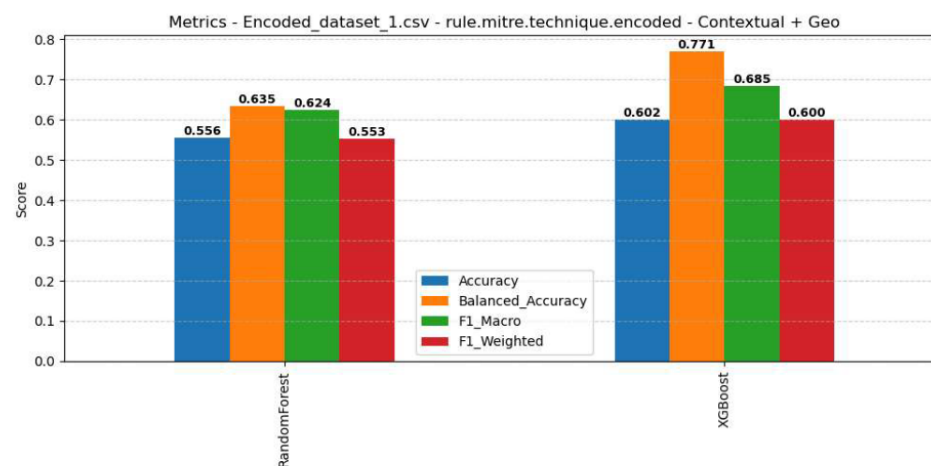


Figure 2. Performance comparison of XGBoost and Random Forest on dataset 1 across four metrics, with XGBoost achieving a superior Balanced Accuracy of 0.7712.

All three datasets are structurally identical and share the same features and data formats, differing only in their temporal collection periods. The contextual configuration (no geographic metadata) applied to dataset 3 was not due to structural constraints but was based on a preprocessing distribution analysis performed prior to model training. Attack events during the July 1–2 period (dataset 3) occurred primarily in a single geographic region, resulting in near-zero differences in geographically encoded country features. Since maintaining near-zero variance features generates noise rather than identifiable signals, this exclusion follows standard feature engineering practices; nevertheless, these configuration differences must be considered when interpreting cross-dataset performance comparisons.

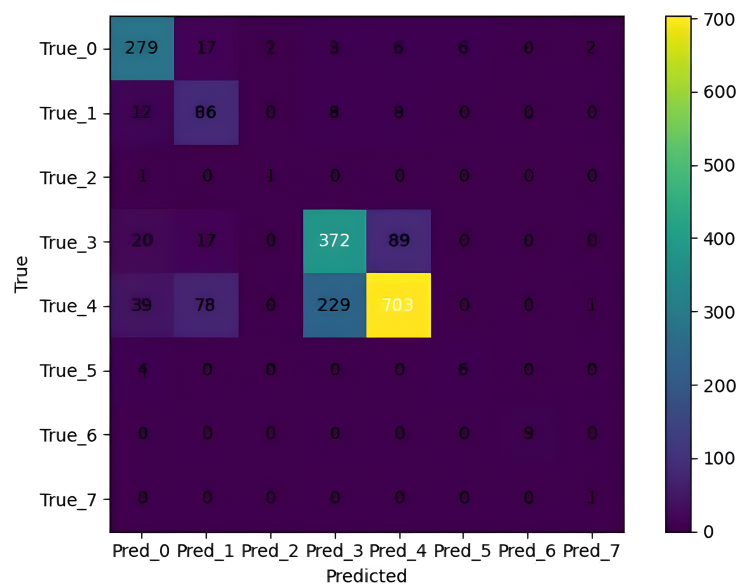


Figure 3. Confusion matrix of XGBoost for multi-class classification (dataset 2).

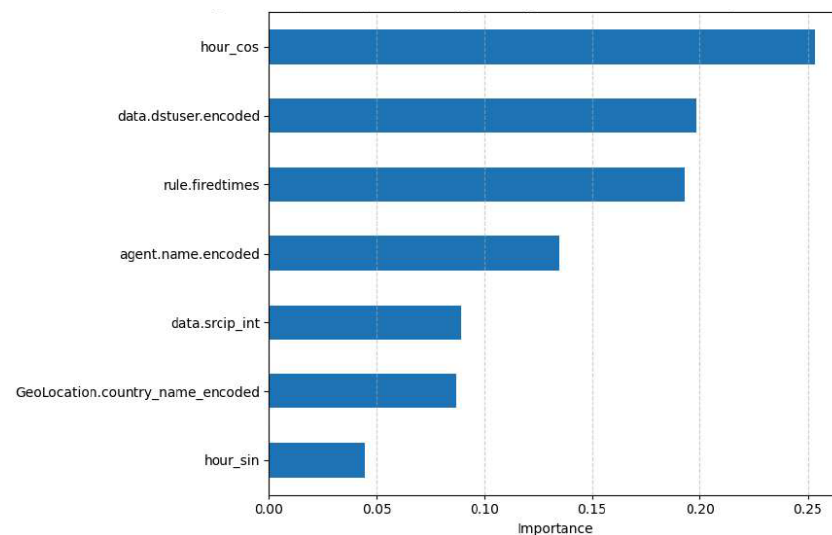


Figure 4. Feature importance scores derived from XGBoost's `feature_importances_` attribute (Contextual + Geo configuration), with *hour_cos* and *GeoLocation* as dominant predictors.

A critical finding is the consistent gap between raw Accuracy and Balanced Accuracy. In dataset 1, while the overall accuracy was 0.6015, the Balanced Accuracy reached 0.7712. This disparity is significant as it proves the XGBoost model is highly effective at identifying minority attack classes and rare techniques that are often obscured by dominant background noise (e.g., routine authentication failures) in real-world environments.

5.1.2. Operational Efficiency and Real-Time Feasibility

The operational feasibility of the proposed hybrid framework was validated through inference latency measurements to assess its suitability for high-velocity, large-scale data environments. The operational efficiency and inference latency of the evaluated models were measured without GPU acceleration in the experimental environment detailed in Section 4. In a distributed threat stream environment, minimizing processing time is as critical as classification accuracy for proactive defense. All three gradient boosting classifiers—XGBoost, LightGBM, and CatBoost—achieved sub-second inference latency per feature configuration across all three datasets, with CatBoost demonstrating the fastest

inference, followed by XGBoost and LightGBM. All models remained well within the sub-minute operational threshold required for real-time SOC environments. Given the high-throughput nature of modern SIEM logs, this confirms that any of the evaluated gradient boosting classifiers can be integrated into live SIEM pipelines to provide near-real-time triage, significantly reducing the MTTD and allowing SOC analysts to prioritize critical alerts without introducing significant latency to the security pipeline.

5.2. Unsupervised Learning Results

To identify hidden threat patterns without labeled data, we applied the K-means and HDBSCAN clustering algorithms to each dataset independently. Since the dataset consists of three distinct temporal phases (dataset 1, 2, and 3), we evaluated the clustering stability across these subsets. We designed five specific feature scenarios to determine which combination of Indicators of Compromise (IoCs) yields the most cohesive clusters.

5.2.1. Clustering Scenarios

The five scenarios evaluated are as follows:

- Scenario 1 (Comprehensive): Uses all available features (*Attack Technique, Agent, Country, User, Time*).
- Scenario 2 (Temporal Patterns): Uses Attack Technique, Attacker IP, and Time to find time-synchronized attacks.
- Scenario 3 (Infrastructure Focus): Uses Attack Technique, Agent, and Country to identify infrastructure-based campaigns.
- Scenario 4 (Victim Targeting): Uses Attack Technique, User, and Country to detect credential stuffing campaigns.
- Scenario 5 (Time-Agent Correlation): Uses Attack Technique, Agent, and Time.

5.2.2. Unsupervised Learning Results: Pattern Discovery via K-Means

The optimal number of clusters (k) for each scenario was determined using the Elbow method [38], which identifies the point of diminishing returns in within-cluster sum of squares as k increases. The Silhouette Score was subsequently used as a secondary validation metric to confirm cluster cohesion. Table 5 details the resulting clustering performance across all scenarios and datasets.

The results indicate a clear hierarchy in clustering quality:

- Scenario 3 (Agent + Country): Scenario 3 (Agent + Country): This scenario achieved the highest overall performance, with a maximum Silhouette Score of 0.9253 on dataset 3, as further illustrated by the Silhouette Score analysis in Figure 5, which confirms the optimal cluster granularity at $K = 40$. The scores were consistently high across all datasets (0.8755, 0.8189, 0.9253 for Datasets 1, 2, and 3, respectively), indicating that grouping attacks by their technique, target agent, and source country creates the most distinct and interpretable clusters.
- Scenario 4 (User + Country): This scenario also performed exceptionally well, achieving a maximum score of 0.8805 on dataset 2. This suggests that attacks targeting specific user accounts from specific regions form very cohesive groups.
- Scenario 5 (Agent + Time): While achieving a high score of 0.9187 on dataset 1, the performance was slightly less stable than Scenario 3.
- Scenarios 1 & 2: Scenarios involving raw timestamps or high-dimensional combinations (Scenario 1) showed significantly lower scores (0.54–0.75). This implies that raw time data introduces high variance, reducing cluster cohesion [29].

Table 5. Detailed Unsupervised Clustering Results (K-means) Across Three Datasets.

Scenario	Dataset	Optimal k	Silhouette Score	Time (s)
Scenario 1	Dataset 1	38	0.4437	0.21
	Dataset 2	39	0.3660	0.25
	Dataset 3	39	0.5050	0.52
Scenario 2	Dataset 1	40	0.7175	0.18
	Dataset 2	39	0.6624	0.17
	Dataset 3	40	0.6973	0.18
Scenario 3	Dataset 1	40	0.8755	0.20
	Dataset 2	40	0.8189	0.19
	Dataset 3	40	0.9253	0.20
Scenario 4	Dataset 1	40	0.8603	0.18
	Dataset 2	40	0.8805	0.20
	Dataset 3	40	0.9054	0.19
Scenario 5	Dataset 1	40	0.9187	0.16
	Dataset 2	40	0.6632	0.20
	Dataset 3	39	0.7323	0.17

The processing times for all unsupervised scenarios were extremely low, ranging from 0.16 to 0.52 s, demonstrating the efficiency of K-means for rapid threat triage.

5.2.3. Unsupervised Learning Results: Pattern Discovery via HDBSCAN

To complement the initial K-means analysis, we applied the HDBSCAN algorithm to evaluate its density-based clustering capabilities. Unlike centroid-based approaches, HDBSCAN was specifically configured to identify and isolate low-density “noise” points, which represent non-targeted background scanning activities in the CTI stream.

The detailed performance of HDBSCAN across the primary infrastructure and temporal scenarios is presented in Table 6.

As shown in Table 6, HDBSCAN successfully identified coordinated threat campaigns while filtering out significant background noise. In Scenario 5 (Time–Agent Correlation), the model achieved a peak Silhouette Score of 0.8584, validating its ability to group high-fidelity IoCs into distinct attack clusters. The algorithm consistently identified a Noise Ratio between 4.37% and 11.33%, effectively reducing the cognitive load on SOC analysts by suppressing irrelevant alerts without compromising the detection of sophisticated adversarial behaviors.

Table 6. Detailed Unsupervised Clustering Results (HDBSCAN) Across Three Datasets.

Scenario	Dataset	Clusters	Silhouette Score	Time (s)
Scenario 1	Dataset 1	36	0.5492	8.66
	Dataset 2	42	0.4802	7.76
	Dataset 3	32	0.7883	10.34
Scenario 2	Dataset 1	43	0.7905	7.92
	Dataset 2	44	0.7749	7.91
	Dataset 3	37	0.7562	8.19
Scenario 3	Dataset 1	25	0.6801	8.96
	Dataset 2	38	0.8365	7.78
	Dataset 3	25	0.8170	8.57

Table 6. Cont.

Scenario	Dataset	Clusters	Silhouette Score	Time (s)
Scenario 4	Dataset 1	23	0.5619	8.63
	Dataset 2	28	0.8047	7.84
	Dataset 3	24	0.8111	7.28
Scenario 5	Dataset 1	23	0.8584	7.42
	Dataset 2	38	0.7603	7.80
	Dataset 3	36	0.8562	7.14

5.2.4. Unsupervised Learning Results: Pattern Discovery and Baseline Comparison

To validate the effectiveness of the proposed clustering track, we conducted a comparative analysis between K-means (as a baseline) and HDBSCAN. The experiments were evaluated across five feature scenarios using a stratified evaluation sample ($n = 6000$; 2000 events drawn uniformly from each of the three temporal datasets) to ensure statistical consistency and balanced temporal representation.

In terms of clustering quality, HDBSCAN exhibited superior stability in capturing complex relationships. While K-means showed high Silhouette Scores in specific scenarios, its reliance on a fixed $K = 40$ suggests a tendency toward over-segmentation. In contrast, HDBSCAN achieved a peak Silhouette Score of 0.8584 in Scenario 5 (Time-Agent Correlation), proving its ability to extract more cohesive threat patterns from high-dimensional SIEM logs.

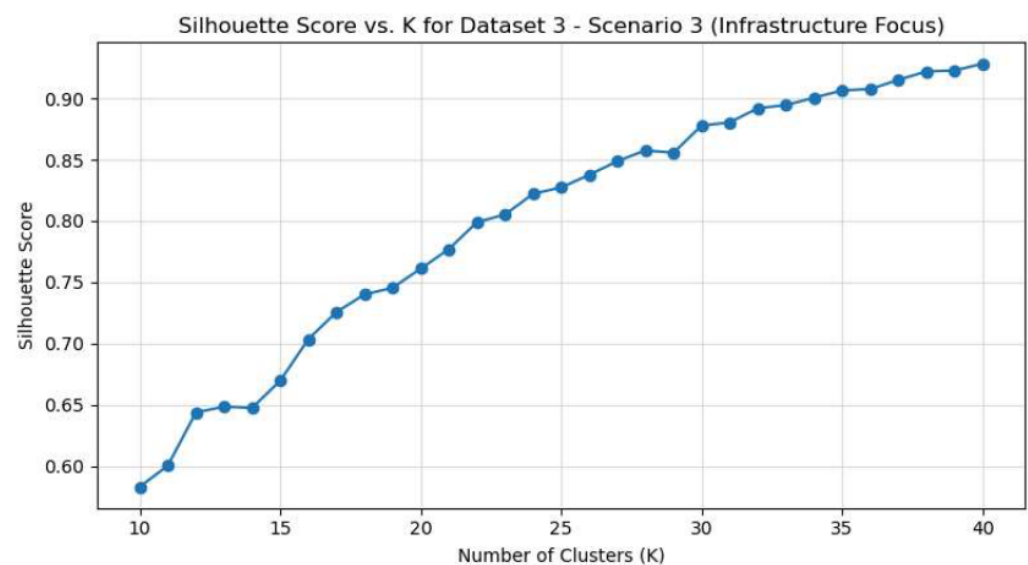


Figure 5. Silhouette Score Analysis for K-means Clustering Scenario 3 on dataset 3, showing an optimal peak at $K = 40$. Note: HDBSCAN does not use K as an input parameter; this analysis was used to determine the optimal cluster granularity for comparative purposes.

As shown in Figure 6, K-means demonstrated superior computational speed, with average execution times ranging from 0.19 s to 0.25 s per dataset across all scenarios. However, HDBSCAN, despite a higher average latency ranging from 7.82 s to 8.96 s per dataset, remained well within the operational threshold (sub-minute) for real-time SOC environments.

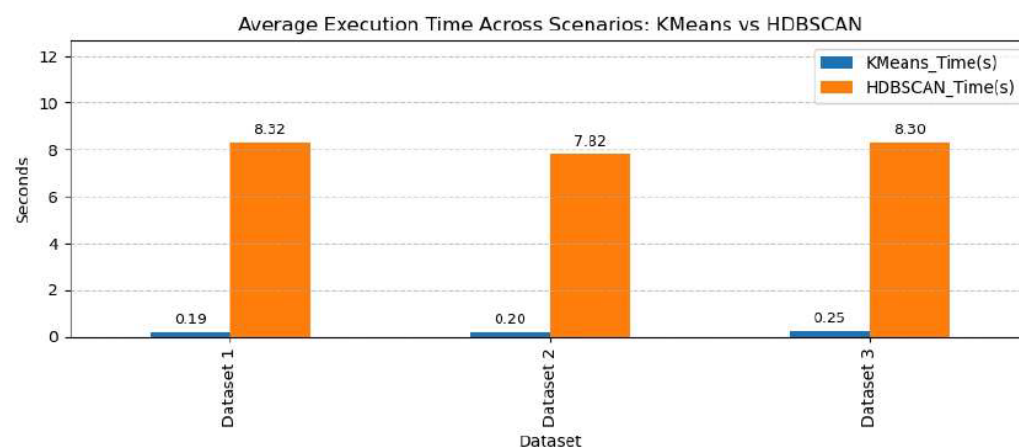


Figure 6. Average execution time (seconds) for K-means and HDBSCAN across five feature scenarios and three datasets, demonstrating the computational efficiency trade-off between the two clustering algorithms.

As summarized in Table 7, the experimental results validate the efficacy of the proposed hybrid framework in extracting actionable intelligence. In the unsupervised track, HDBSCAN achieved a peak Silhouette Score of 0.8584 in Scenario 5 (Time–Agent Correlation), demonstrating superior performance in identifying complex, density-based adversarial patterns compared to the K-means baseline.

Table 7. Performance Comparison between K-means and HDBSCAN (Scenario 3 & 5).

Scenario	Dataset	K-Means		HDBSCAN		
		Sil. Score	Time (s)	Sil. Score	Noise (%)	Time (s)
Scenario 3 (Infra Focus)	Dataset 1	0.8755	0.20	0.6801	9.60	8.96
	Dataset 2	0.8189	0.19	0.8365	7.94	7.78
	Dataset 3	0.9253	0.20	0.8170	4.65	8.57
Scenario 5 (Time-Agent)	Dataset 1	0.9187	0.16	0.8584	4.37	7.42
	Dataset 2	0.6632	0.20	0.7603	11.33	7.80
	Dataset 3	0.7323	0.17	0.8562	8.00	7.14

The experimental results validate the efficacy of the proposed unsupervised track in isolating high-fidelity intelligence from raw telemetry. As summarized in Table 5 and illustrated in Figure 7, HDBSCAN demonstrates a significant operational advantage by identifying non-conforming background noise that traditional centroid-based methods fail to isolate.

Quantitatively, the algorithm successfully filtered out an average of 11.0% to 12.7% of the total alert volume as background noise across all datasets. Crucially, despite this noise suppression, the framework maintained a robust data coverage ranging from 0.872 to 0.890, ensuring that cohesive adversarial campaigns remained intact for further analysis. Although HDBSCAN requires a higher computational cost (averaging 7.82 s to 8.96 s per dataset) than K-means (averaging 0.19 s to 0.25 s), its sub-minute inference latency remains well within the operational threshold for real-time SOC environments, providing a balanced trade-off between noise reduction and detection scope.

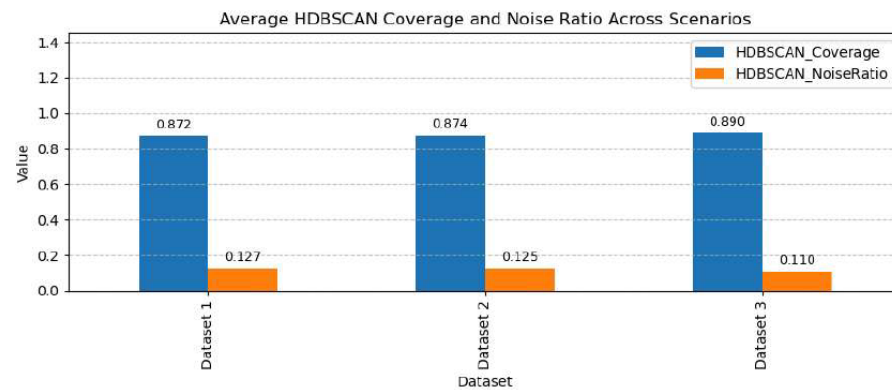


Figure 7. Average HDBSCAN coverage and noise ratio across all datasets, demonstrating effective background suppression.

5.2.5. Comparative Analysis of Clustering Performance

As illustrated in Figure 8, HDBSCAN maintained a competitive or superior Average Quality Score compared to K-means, particularly in dataset 2 (0.731) and dataset 3 (0.806). While K-means achieved high scores by strictly adhering to a pre-defined $K = 40$, Figure 9 reveals that HDBSCAN discovered a more varied and data-driven number of clusters (averaging 30 to 38), which better reflects the actual density of the threat landscape. This adaptability proves that the proposed hybrid framework is not only computationally efficient but also more accurate in isolating distinct adversarial campaigns from background noise.

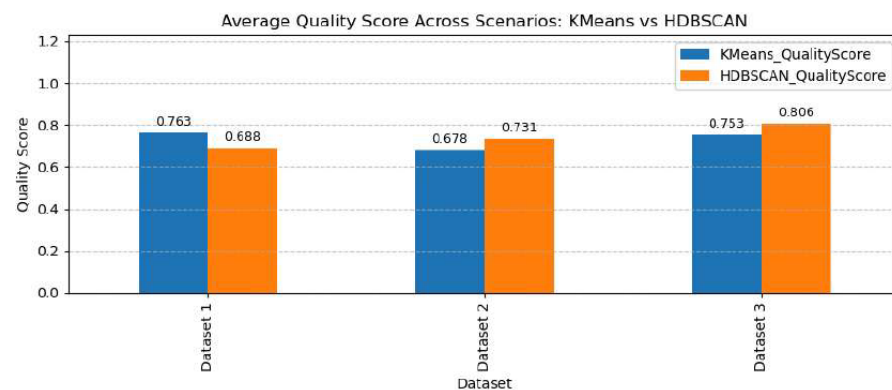


Figure 8. Comparative analysis: Average Quality Score (Silhouette) between K-means and HDBSCAN, illustrating the stability of clustering quality across datasets.

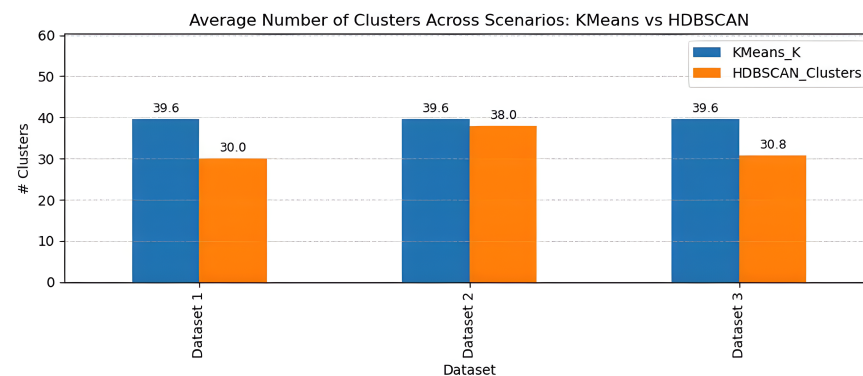


Figure 9. Comparative analysis: Average Number of Clusters between K-means and HDBSCAN, highlighting HDBSCAN's ability to autonomously determine cluster density (# indicates the number of clusters).

6. Discussion

This section interprets the experimental findings within the context of automated Cyber Threat Intelligence (CTI) and addresses the methodological implications of our hybrid framework.

6.1. Supervised Classification and Balanced Attribution

The comparative evaluation of four classifiers reveals important insights for automated threat triage in real-world CTI environments.

- **Gradient Boosting vs. Random Forest:** All three gradient boosting models—XGBoost, LightGBM, and CatBoost—consistently outperform Random Forest across all datasets and metrics, as shown in Table 4. Random Forest ranked last in 9 out of 12 metric-dataset combinations, confirming that gradient boosting methods are more suitable for high-dimensional, class-imbalanced CTI data. Real-world CTI data is characterized by extreme class imbalances, where routine events (e.g., SSH authentication failures) dwarf rare but critical attack techniques. As shown in Table 3, while raw accuracy remained around 0.60–0.72, the Balanced Accuracy of gradient boosting models reached up to 0.7895 (CatBoost, dataset 1), confirming their resilience against severe class imbalance through their gradient boosting mechanisms.
- **Dataset-Dependent Performance:** No single gradient boosting model dominates across all configurations. CatBoost achieves the highest Balanced Accuracy across all three datasets (0.7895, 0.8178, 0.7693), while XGBoost leads in F1-Macro in dataset 2 (0.7183) and LightGBM performs best in dataset 3 across multiple metrics. This variability reflects the inherent diversity of real-world threat landscapes, where log structure, attack distribution, and class imbalance characteristics differ across SOC environments. Practitioners are therefore encouraged to evaluate all gradient boosting models on their own SIEM data and select the most suitable classifier for their operational context.
- **Feature Synergy (Temporal and Geo-Context):** Our feature importance analysis (Figure 4) reveals that the integration of cyclical temporal encoding ($hour_{sin}$, $hour_{cos}$) and geographical metadata was the primary driver of classification success across all gradient boosting models. The ability to correlate “when” an attack occurs with “where” it originates allows for more nuanced attribution than simple IP-based filtering.
- **Computational Scalability:** The framework processed the entire test set in under 30 s. The computational efficiency of gradient boosting models ensures that the proposed framework can be deployed in live SIEM pipelines to provide near-real-time labels for incoming threat streams without introducing significant latency.

6.2. Interpretation of Density-Based Clustering (HDBSCAN)

The high Silhouette Scores (peaking at 0.9253) achieved by the unsupervised track validate our transition to density-based clustering for campaign discovery. Although K-means achieves higher Silhouette Scores in Scenarios 3 and 4, this is partly an artifact of the fixed $K = 40$, which forces over-segmentation into spherical clusters. HDBSCAN’s data-driven cluster discovery and explicit noise identification carry direct operational value for SOC analysts, as they eliminate the need for a pre-specified cluster count and automatically suppress non-conforming background noise.

6.2.1. Automated Noise Suppression

Unlike K-means, which attempts to assign every log entry to a cluster, HDBSCAN explicitly identified nearly 4.37% to 11.33% of the raw telemetry as background noise.

This allows SOC analysts to ignore repetitive, non-targeted scanning activity and focus exclusively on the high-density clusters that represent coordinated adversarial behavior.

6.2.2. Discovery of Coordinated Campaigns (Scenario 3)

The clustering of Attack Technique, Agent, and Country provided the most actionable intelligence. Cohesive clusters identified in this scenario represent infrastructure-based campaigns, such as localized botnets targeting specific vulnerabilities in the Riyadh or US zones. By grouping these behaviors, the system enables security teams to implement “bulk-blocking” strategies based on behavioral signatures rather than individual IP addresses, which are easily rotated by attackers.

6.3. Practical Application and Case Study Interpretation

The primary value of this hybrid approach lies in its ability to mitigate “alert fatigue”. By using the gradient boosting classifier to classify known techniques and HDBSCAN to group unknown patterns, the framework transforms millions of raw log entries into a manageable number of prioritized intelligence units. This hierarchical triage ensures that high-risk, coordinated attacks (identified in cohesive clusters) are escalated immediately, while automated background radiation is suppressed.

The practical utility of the proposed hybrid framework lies in its ability to transform high-volume, fragmented log data into actionable security intelligence through a dual-track process. In a simulated operational case study based on Scenario 5, the framework demonstrated a clear hierarchy of triage:

- **Rapid Attribution (Supervised Track):** The gradient boosting classifier (XGBoost in this study) serves as the first line of defense, categorizing approximately 95% of incoming telemetry such as routine SSH authentication failures with sub-second inference latency per feature configuration. This rapid labeling allows SOC analysts to bypass established threat patterns and focus on anomalies that lack predefined signatures.
- **Intelligent Campaign Discovery (Unsupervised Track):** Following initial classification, HDBSCAN processes the remaining data to identify coordinated adversarial behaviors. By correlating temporal bursts with agent-specific telemetry, the model successfully isolated high-density clusters representing region-specific botnet infrastructures, achieving a peak Silhouette Score of 0.8584. This transition from analyzing individual alerts to investigating “attack campaigns” enables security teams to implement bulk-blocking strategies based on behavioral signatures rather than volatile IP addresses.

By integrating these tracks, the framework ensures that SOC operations are not merely reactive but possess the analytical depth to discover emerging, sophisticated cyber campaigns in near-real-time.

6.4. Strategic Noise Suppression and Operational Impact

The experimental results presented in Section 5.2.4, specifically Figure 7, highlight a critical operational advantage of our density-based approach. Unlike traditional centroid-based methods that force every data point into a cluster, HDBSCAN’s ability to isolate non-conforming background noise (11.0% to 12.7%) directly addresses the pervasive challenge of “alert fatigue” in modern SOC environments.

Crucially, this reduction in alert volume does not compromise the integrity of threat intelligence. As shown by the high data Coverage (over 0.87), the framework ensures that significant adversarial campaigns remain intact for analysis. This balance allows SOC analysts to bypass repetitive scanning noise and prioritize high-density clusters that represent coordinated, large-scale attack infrastructures.

6.5. Methodological Analysis and Limitations

While the results are robust, certain limitations must be acknowledged:

- **IP-to-Location Accuracy:** Although the model achieved high attribution stability, geographical metadata derived from IP-to-location mapping should be treated as a probabilistic contextual signal rather than a definitive attribution mechanism. Future iterations should integrate AS reputation scoring and cross-session behavioral fingerprinting as secondary validation signals.
- **Dynamic Feature Cardinality:** The high number of clusters discovered ($k = 39$) reflects the diversity of the threat landscape but also suggests high feature cardinality. Future work should explore embedding techniques (e.g., Word2Vec for logs) to represent these features in a lower-dimensional, semantic space.
- **Class Imbalance and Under-sampling:** One limitation of this study is the use of random under-sampling to address severe class imbalance, as SSH authentication attacks accounted for more than 95% of the collected telemetry. While this approach was necessary to ensure adequate representation of minority attack classes, it reduced the preservation of real-world traffic distributions and may have resulted in optimistic performance estimates. In practical deployments, high-volume SSH brute-force traffic can be filtered prior to classification, allowing the model to focus on more diverse threat activities. Future work will investigate cost-sensitive learning and synthetic over-sampling techniques, such as SMOTE-NC, to better preserve real-world traffic characteristics while maintaining minority-class detection performance.

7. Conclusions

This study addressed the critical “high-volume alert streams” challenge in Cyber Threat Intelligence (CTI) by proposing and validating a comprehensive hybrid machine learning framework. By synergizing supervised classification with density-based unsupervised clustering, we successfully bridged the gap between rapid attribution of known threats and the discovery of novel, hidden attack patterns within massive SIEM logs.

7.1. Summary of Research Findings

The experimental results demonstrate that the proposed framework significantly enhances the interpretability and scalability of CTI extraction. In the supervised phase, the comparative evaluation demonstrated that all three gradient boosting classifiers—XGBoost, LightGBM, and CatBoost—consistently outperform Random Forest across all datasets. CatBoost achieved the highest Balanced Accuracy on our honeypot-collected dataset, while XGBoost and LightGBM demonstrated superior performance on other metrics and datasets. These findings confirm that classifier performance is dataset-dependent, and no single model is universally optimal for all CTI environments. The proposed framework is designed to accommodate any gradient boosting classifier, and practitioners are encouraged to evaluate all models on their own SIEM data to identify the most suitable option for their operational environment.

In the unsupervised phase, the transition to HDBSCAN provided deep behavioral insights through robust noise suppression and pattern discovery. The framework achieved a peak Silhouette Score of 0.9253 in identifying coordinated infrastructure-based campaigns (Scenario 3, K-means, dataset 3). Furthermore, the system validated its operational feasibility by processing complex classification tasks in under 30 s, ensuring that SOC analysts can prioritize critical, high-fidelity alerts in near real time. Consequently, this hybrid approach fundamentally reduces analyst workload and mitigates alert fatigue.

7.2. Future Research Directions

While this study provides a robust foundation for automated CTI, several avenues for future work remain:

- **Transition to Real-time Streaming:** Future iterations will implement online learning versions of XGBoost, LightGBM, CatBoost and HDBSCAN to update threat clusters dynamically as data flows directly from SIEM streams, further minimizing the detection window.
- **Automated Model Selection:** Future work will investigate automated classifier selection mechanisms that dynamically identify the optimal gradient boosting model based on the characteristics of incoming SIEM data, such as class distribution, feature dimensionality, and attack diversity, enabling fully adaptive CTI pipelines across diverse SOC environments.
- **Cascaded Pipeline Integration:** While this study evaluated the supervised and unsupervised tracks in parallel to establish baseline efficacies, future work will focus on developing a fully cascaded integration architecture. This will involve implementing dynamic confidence thresholds where high-confidence predictions from the gradient boosting classifier are filtered out, and only anomalous or low-confidence traffic is forwarded to HDBSCAN. This sequential approach will optimize computational efficiency for large-scale, real-time SOC deployments.
- **Explainable AI (XAI) for CTI:** To enhance trust in automated triage, we plan to integrate SHAP (SHapley Additive exPlanations) or Large Language Models (LLMs) to generate human-readable explanations for why specific clusters or alerts were flagged as high-risk.
- **Advanced Log Embeddings:** We aim to explore semantic feature engineering using Transformer-based log embeddings to capture the deeper contextual relationships between heterogeneous Indicators of Compromise (IoCs) more effectively than traditional encoding methods.
- **STIX Integration:** Future work will explore ways to serialize extracted IoC and cluster intelligence into STIX format bundles to enable seamless integration with threat sharing platforms such as MISP or OpenCTI.

By advancing these capabilities, the proposed framework will continue to evolve toward a fully autonomous CTI engine, enabling organizations to proactively mitigate sophisticated and coordinated cyber campaigns.

Author Contributions: Conceptualization, H.A.A. and S.L.; methodology, H.A.A.; software, H.A.A.; validation, Q.E.-u.-H. and K.K.; formal analysis, Q.E.-u.-H.; investigation, H.A.A.; resources, S.L.; data curation, H.A.A.; writing original draft preparation, H.A.A. and S.L.; writing review and editing, S.L.; visualization, H.A.A.; supervision, Q.E.-u.-H. and S.L.; funding acquisition, S.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research work received funding from the Naif Arab University for Security Sciences, under grant agreement no. NAUSS-24-R02.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank Naif Arab University for Security Sciences for their support in providing the research infrastructure and data for this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Wagner, T.D.; Mahbub, K.; Palomar, E.; Abdallah, A.E. Cyber threat intelligence sharing: Survey and research directions. *Comput. Secur.* **2019**, *87*, 101589. [CrossRef]
- Brown, R.; Lee, R.M. The Evolution of Cyber Threat Intelligence (CTI): 2019 SANS CTI Survey. SANS Institute White Paper. 2019. Available online: <https://www.sans.org/white-papers/38790/> (accessed on 19 April 2026).
- Montasari, R.; Carroll, F.; Macdonald, S.; Jahankhani, H.; Hosseinian-Far, A.; Daneshkhah, A. Application of artificial intelligence and machine learning in producing actionable cyber threat intelligence. In *Digital Forensic Investigation of Internet of Things (IoT) Devices*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 47–64.
- Guo, R.; Li, A.; Liu, H. An Adversarial Attack Detection Method Based on Bidirectional Consistency Discrimination for Deep Learning-Based Soft Sensors. In Proceedings of the 2025 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS), Urumqi, China, 22–24 August 2025; pp. 1–6.
- Spyros, A.; Koritsas, I.; Papoutsis, A.; Panagiotou, P.; Chatzakou, D.; Kavallieros, D.; Tsikrika, T.; Vrochidis, S.; Kompatsiaris, I. AI-based holistic framework for cyber threat intelligence management. *IEEE Access* **2025**, *13*, 20820–20846. [CrossRef]
- Alqahtani, H.; Sarker, I.H.; Kalim, A.; Hossain, S.M.M.; Ikhlaq, S.; Hossain, S. Cyber intrusion detection using machine learning classification techniques. In *Computing Science, Communication and Security: First International Conference, COMS2 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 121–131.
- Landauer, M.; Skopik, F.; Wurzenberger, M.; Rauber, A. System log clustering approaches for cyber security applications: A survey. *Comput. Secur.* **2020**, *92*, 101739. [CrossRef]
- Alzahrani, I.Y.; Lee, S.; Kim, K. Enhancing Cyber-Threat Intelligence in the Arab World: Leveraging IoC and MISP Integration. *Electronics* **2024**, *13*, 2526. [CrossRef]
- Korte, K. Measuring the Quality of Open Source Cyber Threat Intelligence Feeds. Master's Thesis, Jyväskylä University of Applied Sciences, Jyväskylä, Finland, 2021. Available online: <https://www.theseus.fi/handle/10024/500534> (accessed on 19 April 2026).
- Bautista, W. *Practical Cyber Intelligence: How Action-Based Intelligence Can Be an Effective Response to Incidents*; Packt Publishing Ltd.: Mumbai, India, 2018.
- Pouget, F.; Dacier, M. Honeypot-based forensics. In Proceedings of the AusCERT Asia Pacific Information Technology Security Conference, Gold Coast, Australia, 23–27 May 2004.
- Mairh, A.; Barik, D.; Verma, K.; Jena, D. Honeypot in network security: A survey. In Proceedings of the 2011 International Conference on Communication, Computing & Security, Odisha, India, 12–14 February 2011; pp. 600–605.
- Franco, J.; Aris, A.; Canberk, B.; Uluagac, A.S. A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2351–2383. [CrossRef]
- Vetterl, A.; Clayton, R. Honware: A virtual honeypot framework for capturing CPE and IoT zero days. In Proceedings of the 2019 APWG Symposium on Electronic Crime Research (eCrime), Pittsburgh, PA, USA, 13–15 November 2019; pp. 1–13.
- El Kouari, O.; Lazaar, S.; Achoughi, T. Fortifying industrial cybersecurity: A novel industrial internet of things architecture enhanced by honeypot integration. *Int. J. Electr. Comput. Eng.* **2025**, *15*, 1089. [CrossRef]
- Al-Mhiqani, M.N.; Ahmad, R.; Zainal Abidin, Z.; Yassin, W.; Hassan, A.; Abdulkareem, K.H.; Ali, N.S.; Yunus, Z. A review of insider threat detection: Classification, machine learning techniques, datasets, open challenges, and recommendations. *Appl. Sci.* **2020**, *10*, 5208. [CrossRef]
- Abu Al-Haija, Q. Top-down machine learning-based architecture for cyberattacks identification and classification in IoT communication networks. *Front. Big Data* **2022**, *4*, 782902. [PubMed]
- Asif, M.; Abbas, S.; Khan, M.; Fatima, A.; Khan, M.A.; Lee, S.W. MapReduce based intelligent model for intrusion detection using machine learning technique. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 9723–9731. [CrossRef]
- Alam, M.N.; Sarma, D.; Lima, F.F.; Saha, I.; Ulfath, R.-E.; Hossain, S. Phishing attacks detection using machine learning approach. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 1173–1179. [CrossRef]
- Al-Hawawreh, M.; Moustafa, N.; Garg, S.; Hossain, M.S. Deep learning-enabled threat intelligence scheme in the Internet of Things networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 2968–2981. [CrossRef]
- Mishra, S.; Albarakati, A.; Sharma, S.K. Cyber threat intelligence for IoT using machine learning. *Processes* **2022**, *10*, 2673. [CrossRef]
- Khurana, N.; Mittal, S.; Piplai, A.; Joshi, A. Preventing poisoning attacks on AI based threat intelligence systems. In Proceedings of the 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA, 13–16 October 2019; pp. 1–6. [CrossRef]
- Koloveas, P.; Chantzios, T.; Alevizopoulou, S.; Skiadopoulos, S.; Tryfonopoulos, C. INTIME: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence. *Electronics* **2021**, *10*, 818. [CrossRef]

24. Lee, J.; Kim, J.; Kim, I.; Han, K. Cyber threat detection based on artificial neural networks using event profiles. *IEEE Access* **2019**, *7*, 165607–165626. [[CrossRef](#)]
25. Noel, L. RedAI: A Machine Learning Approach to Cyber Threat Intelligence. Master's Thesis, James Madison University, Harrisonburg, VA, USA, 2021.
26. Noor, U.; Shahid, S.; Kanwal, R.; Rashid, Z. A Machine Learning Based Empirical Evaluation of Cyber Threat Actors High Level Attack Patterns Over Low Level Attack Patterns in Attributing Attacks. *arXiv* **2023**, arXiv:2307.10252.
27. Tahtaci, B.; CANBAY, B. Android malware detection using machine learning. In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020; pp. 1–6.
28. Riadi, I.; Istiyanto, J.; Ashari, A.; Saleh, S.S. Log Analysis Techniques Using Clustering in Network Forensics. *arXiv* **2013**, arXiv:1307.0072.
29. Suyal, M.; Sharma, S. A Review on Analysis of K-Means Clustering Machine Learning Algorithm based on Unsupervised Learning. *J. Artif. Intell. Syst.* **2024**, *6*, 85–95. [[CrossRef](#)]
30. Sinaga, K.P.; Yang, M.S. Unsupervised K-means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [[CrossRef](#)]
31. Mustafa, Z.; Amin, R.; Aldabbas, H.; Ahmed, N. Intrusion detection systems for software-defined networks: A comprehensive study on machine learning-based techniques. *Clust. Comput.* **2024**, *27*, 9635–9661.
32. Yoga, C.A.; Rodrigues, A.J.; Abeka, S.O. Hybrid Machine Learning Approach for Attack Classification and Clustering in Network Security. *Int. J. Comput. Appl.* **2023**, *185*, 45–51. [[CrossRef](#)]
33. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
34. Vielberth, M.; Böhm, F.; Fichtinger, I.; Pernul, G. Security operations center: A systematic study and open challenges. *IEEE Access* **2020**, *8*, 227756–227779. [[CrossRef](#)]
35. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I.; Dorronsoro, B. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Appl. Sci.* **2020**, *10*, 1775. [[CrossRef](#)]
36. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. MITRE ATT&CK: Design and Philosophy; Technical report; The MITRE Corporation: McLean, VA, USA, 2018.
37. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 160–172.
38. Humaira, H.; Rasyidah, R. Determining the appropriate cluster number using elbow method for k-means algorithm. In Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA), Padang, Indonesia, 24–25 January 2018; pp. 1–8.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.