future internet

MDPI

Article

Threat Actor Attribution Applying a Tactics–Techniques–Procedures Approach: An Empirical Investigation

Shaheen Hussain and Krassie Petrova *

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; shaheen.hussain404@gmail.com

* Correspondence: krassie.petrova@aut.ac.nz

Abstract

The increasing frequency and growing impact of cyberattacks have led organizations to adopt proactive defense approaches to cybersecurity risk mitigation, especially in the case of advanced persistent threats (APTs). The correct identification of the specific malicious actors behind a cyberattack is important for the success of incident response and for the investigative work of the security operations center (SOC) team. This research explores the capabilities and limitations of a machine learning (ML) approach to identifying malicious actors and the threats they pose (threat actor attribution) based on the tactics, techniques, and procedures (TTP) observed in specific cybersecurity incidents and on the incident context (the geographical location and industry affiliation of the victims targeted in the attack). A large language model (LLM) was used to extract TTPs from the MITRE ATT&CK database of cybersecurity incidents. The experiments included modeling threat actor attribution using five ML algorithms: k-nearest neighbors (KNN), decision tree (DT), random forest (RF), support vector machine (SVM), and naïve Bayes (NB), with different methods applied for feature selection and weighting. The results indicated that model accuracy and other performance metrics were significantly improved when the input dataset included both TTP and contextual features. The KNN and SVM models produced the best performance results; the highest classification accuracy achieved was 93.19%. The outcomes of this study may be applied by cybersecurity professionals to identify malicious actors, estimate the number and types of data points that are required to adequately attribute a cyberattack to an actor, and improve the accuracy of the classification by weighting the input dataset features.

Keywords: threat actor attribution; advanced persistent threats; tactics; techniques and procedures; APT; TTP; machine learning; ML; KNN; SVM; RF; DT; cybersecurity; cyberattack; cybersecurity incident

Academic Editors: Marcelo García and Paulina E. Ayala

Received: 6 November 2025
Revised: 3 August 2026
Accepted: 8 August 2026
Published: 13 August 2026

1. Introduction

Due to the growing global cyberattack surface comprising organizations and individual entities, the risk and complexity of the attacks have increased significantly [1]. The most prevalent cyberattacks include denial-of-service (DoS) and malware attacks, phishing, zero-day exploitation, and the identification and exploitation of unpatched vulnerabilities. Investigations of current trends indicate that the frequency and potential impact of these attacks have been increasing over recent years [2,3], while the techniques used by malicious actors to launch cyberattacks are constantly evolving to improve their success rates [4].

A current trend in proactive defense approaches is the use of threat intelligence as a means of collecting and sharing information relating to cyberattacks, including the specific attack indicators and techniques observed; such actionable intelligence is commonly shared through threat reports by security vendors [5]. This information may be used to identify the malicious person or group behind a cyberattack, or the 'threat actor' [6].

Threat actor attribution is a cybersecurity defense approach that involves linking a cybersecurity attack to a specific threat actor [7]. The process involves the analysis of cyber incident technical characteristics such as the IP address of the attack source, and behavioral characteristics such as the tactics, techniques, and procedures (TTP) used by the threat actor.

In larger organizations, the protection of the digital assets of the organization is managed by the security operations center (SOC) team [8]. With the increasingly complex and polymorphic cyber threat landscape, it is important to understand how malicious actors operate and use the knowledge gained to inform defense strategies and tactics [9]. Therefore, accurate threat actor attribution is critical for ensuring the efficacy and increasing the efficiency of the cybersecurity response [10].

The MITRE ATT&CK Framework [11,12] is one of the most commonly used formats for capturing threat actor TTPs. The standardized framework format used in the MITRE ATT&CK Database [13] facilitates information exchange; it enables correlating and analyzing cyber incident indicators such as TTPs to derive relevant and accurate contextual intelligence. Understanding the nature of the cyberattack and how it is tailored to the targeted victim's location and industry sector affiliation helps SOC teams accumulate and codify knowledge about the current threat landscape to enable and support the protection of the organization's digital assets [14].

1.1. Research Background

1.1.1. Cyberattacks and Threat Intelligence

As a result of the evolving cyber threat landscape, significant cybersecurity incidents (cyber incidents) experienced by governments, multinational corporations, and small businesses continue to occur. Different categories of attackers are involved, including state-sponsored threat actors, financially motivated hacker groups, and hacktivists.

Attacker motivation varies from espionage to financial gain, or even causing damage and harm to human life [15–20]. For example, Chng et al. [16] identify financial profit and revenge as the most influential of the eight motivation factors driving hacker activity. Similarly, Alqudhaibi et al. [17] found that espionage followed by destruction, hacktivism, financial gain, and revenge were the five distinct motives behind cyberattacks on critical infrastructure.

Espionage and financial gain continue to motivate cybercriminals and nation-state actors such as LAPSUS$ and APT 29 (Cozy Bear), who have launched a series of successful cyberattacks [18]. Consequently, the financial impact of cybercrime has increased significantly. According to Masmoudi et al.'s [19] projections, cybercrime costs will reach US$ 23.82 billion in 2027 (while in 2018 it was still well below US$ 1 billion). The authors point out that the diversity and complexity of the cyberattacks have increased as well. While threat actors may still be driven by multiple intrinsic and extrinsic motivational factors, information theft, espionage, and financial profit are currently the most powerful motives for attacks on critical infrastructure (e.g., energy, manufacturing, the gas and oil industry) [20]. The most sophisticated cyberattacks, motivated by information theft and espionage, are launched by well-funded state actors; these are closely followed by highly organized criminal groups whose motivation is predominantly financial, e.g., profiting through ransomware attacks [21–23].

In the organization, the security operations center (SOC) team is tasked with the deployment of security solutions to protect the company's network, endpoint devices, and enterprise-wide applications. In addition to identifying, analyzing, and responding to cybersecurity incidents, the SOC team needs to ensure compliance with the relevant national and international regulatory requirements, such as the Health Insurance Portability and Accountability Act (HIPAA), the General Data Protection Regulation (GDPR), and the Payment Card Industry Data Security Standard (PCI DSS) [24]. Adherence to standards (e.g., ISO/IEC27001) and best industry practices is also an aspect of security compliance [25].

The focus of SOC team operations may differ across industry sectors. According to Govea et al. [26], critical infrastructure systems are characterized by a high level of interconnectedness and reliance on digital technologies. This makes them particularly vulnerable to long-term, hidden cyber threats known as advanced persistent threats (APTs) that target industrial control centers (ICCs). In manufacturing environments, the longer cycle of updating equipment and operational procedures may lead to outdated technology and processes that are widely exposed to cyberattacks [17]. In the healthcare sector, information theft attacks and ransomware attacks threaten patient data privacy and patient well-being [27].

SOC team experts use a variety of third-party tools, including AI agents, to collect and manage threat intelligence data related to cyber incidents, and to investigate alerts and suspicious activities that may be signaling a cyberattack [28,29]. For example, security information and event management (SIEM) systems provide a centralized platform to perform SOC operations [30]; threat hunting tools are used to search for malicious penetrations that have bypassed the security defense, while digital forensics techniques aid in discovering the cause of cybersecurity incidents [28]. To cope with the complexity of its resource-demanding tasks, SOC teams adopt AI-based platforms such as CrowdStrike to assist in the analysis of the large data volumes to identify potential attack patterns and support cybersecurity incident response [31]. The establishment of a collaborative AI-human SOC team has been proposed as a means of increasing SOC operations efficiency against the backdrop of a fast-evolving cyber threat landscape and increased alarm levels [32].

1.1.2. Threat Actor Attribution

Identifying and understanding the cyberattack tactics and techniques used by threat actors has demonstrated value in improving the design of cybersecurity defense systems [33]. Understanding threat actors' techniques can enhance an organization's ability to detect threats and enable a proactive approach to providing protection against emerging threats [34]. However, the threat actor attribution process of identifying the perpetrator behind a cyberattack is a complex task that may be hard to accomplish. It involves the analysis of pertinent technical data such as IOC and TTP digital footprints, as well as a variety of supporting information, for example, a targeted victim profile including the relevant industry sector and their geographical location [35]. Even if only probable and not completely reliable, threat actor attribution results equip organizations with valuable knowledge about their most relevant adversaries and threats [36]. By attributing a cyberattack to a threat actor, organizations gain the ability to deter threats by counteracting the relevant TTPs and reducing cyberattack success rates. Additionally, threat actor attribution allows SOC teams to prioritize their response to incidents [20,37].

The MITRE ATT&CK Framework (developed in 2013) classifies the methods and approaches used in cyberattacks by identifying attack TTPs (the activities used at the different stages of an attack) [12]. Currently, the database comprises a total of 203 techniques and 453 sub-techniques [38]. Another source of threat intelligence is indicators of compromise

(IOCs), also shared by commercial security vendors and industry- and government-based agencies. IOCs provide IP addresses, suspicious file hashes, and domain names that are often associated with specific threat actors [39]. However, IOCs are less effective than TTPs as an approach to improving the effectiveness of the organization's security controls against attacks. While attackers can change their IOCs by using a different IP address or a URL or editing a file to change its hash value, TTPs are often more difficult to alter [40,41]. Gopalakrishna et al. [42] place TTPs at the top of the tactical layer of their 'Extended Pyramid of Pain', highlighting their cybersecurity defense potential.

1.2. Related Research Work

1.2.1. Cybersecurity Data Generation Approaches

The MITRE ATT&CK Framework provides security analysts with the capability to decipher attacks in relation to a comprehensive and standardized catalog of TTPs. Bhole et al. [20] explored the datasets related to threat actor activities through threat intelligence and real-world incidents that were used to analyze cyber threat groups. The public databases ThaiCERT, Malpedia, ICS-CERT, and the European Repository of Cyber Incidents (EuRepoC) are compatible with the format of the TTPs in MITRE ATT&CK and have been used in a number of empirical investigations. However, prior research has also demonstrated that the use of heterogeneous cyber intelligence data sources and hybrid feature sets enhances threat actor attribution [43]. For example, Chen et al. [44] successfully extracted TTPs from threat intelligence reports and used them for APT group clustering based on their attributes and techniques.

To mitigate the challenges posed by manual TTP extraction, Domschot et al. [45] compared various machine learning (ML) techniques to associate specific words with different TTPs and extract relevant TTPs directly from intelligence reports; they achieved an accuracy of 69.8%. In a related study, Permana et al. [46] attempted to extract TTPs from unstructured data (cyber threat news reports) by parsing and searching for selected matching words. They formatted the information gathered using structured threat information expression (STIX). The STIX format is machine-readable and is supported by a number of industry and community platforms [47]. Huang et al. [48] also aimed to address the deficiencies of manual extraction of TTPs. They developed a deep learning (DL) model for the efficient extraction and classification of MITRE techniques from unstructured cyber threat intelligence (CTI) reports.

DL-based generative artificial intelligence (GenAI) models can extract attack patterns and indicators of attack from threat intelligence data to help cybersecurity teams [49]. For example, Saddi et al. [50] proposed a conceptual GenAI model for automated threat intelligence gathering, updating, and analysis. GenAI applications such as a large language model (LLM) can be an effective tool for identifying and extracting TTPs from threat intelligence reports, as shown by Fieblinger et al. [5]. For example, MITRE has enhanced the accuracy of their ATT&CK mapper (TRAM) by deploying an LLM [51]. The use of the commercially available ChatGPT (an LLM) as a tool to analyze security-related information from a variety of sources and identify behaviors and attack patterns has been suggested in more recent research [52,53].

1.2.2. Prompt Engineering for LLMs

The performance of LLM models is sensitive to the quality of the 'prompting' (structured input provided by the user). By applying prompt engineering (PE) best practices, the user can improve model performance, increase the accuracy of the response, and overall, gain greater control over the process [54]. For example, the LLM can be instructed to

double-check its output and to re-attempt the task if there are any inconsistencies [44]. Such PE techniques aim to increase the adequacy and the credibility of the LLM's responses.

Schmidt et al. [55] recently found that prompt model performance improves when several prompt components are included; this ensures that the LLM has sufficient information to respond with optimized and accurate results. Providing comprehensive prompts means building a strong instructional foundation for the LLM to follow. In the flipped interaction pattern, the user instructs the LLM to ask questions (which the user replies to) and continue till it has enough information to perform the required action; in the reflection pattern, the user asks the LLM to provide supporting rationale on how and why it responded with the answer to the prompt. Furthermore, it is suggested by Tripaldelli et al. [56] that specifying the role of the LLM (e.g., by stating in the prompt that the LLM is an expert on the subject), or including step-by-step instructions on how the LLM should perform its task and the specific output required, also results in improved responses to prompts.

1.2.3. Approaches to Threat Actor Attribution

The threat actor attribution process is a time-consuming and challenging task if conducted manually [57]. Noor et al. [58] suggest that ML modeling for their analysis and attribution to criminal identities based on a dataset of the features of the TTPs extracted from threat intelligence can produce useful and reliable results to support cyberattack investigation. However, research in actor and threat attribution is relatively scarce. Abou et al. [18] and Desa et al. [59] have examined threat actor typologies and classifications, motivations and behaviors, and have noted that there is still a need to develop reliable methods that can enhance threat actor profiling and cybersecurity management and best practices.

Data extracted from cyberthreat technical descriptions such as the TTPs and the low- and high-level IOCs in David Bianco's 'Pyramid of Pain' framework [60] have been extensively used in current empirical research [44,61–66]. This makes it possible to utilize the collective cybersecurity intelligence knowledge and also to align threat actor profiling with cybersecurity practitioners' expectations [67]. Several studies have reported high accuracy levels.

For example, Duan et al. [61] gathered indicators ranging from low-level IOCs, such as IP addresses and domain names, to higher-level indicators, including malware and TTPs. For attribution, their study used the information about the country and the industry sector targeted by the attack. The authors identified the belongs-to relationship between their dataset of threat actors and a corresponding incident dataset, in which they utilized a heterogeneous graph technique for mapping nodes to threat actors. The dataset comprised 1027 threat actor nodes; an 80/20 split was used to train, validate, and test the model. The authors compared the performance of eight different learning models against their dataset; the highest precision model achieved 54.42% accuracy in positioning the threat actor within the top five most probable threat actors behind an incident.

Kida et al. [62] explored threat actor attribution with regard to malware that was used in attacks and applied a range of classification algorithms to match an attack to a threat actor: random forest (RF), support vector machine (SVM), naïve Bayes (NB), and k-means clustering. The authors used a sample of 3594 malware file hashes mapped to 12 threat actor groups (with an 80/20 split for training and testing). The best-performing model achieved 89% accuracy in associating malware hashes with threat actors.

Xiao et al. [63] used IOCs, malware, and TTPs to derive the data points needed for threat actor attribution modeling. The authors developed a baseline for matching threat actors to threat intelligence reports using multilevel heterogeneous graphs and utilized data from 1300 threat intelligence reports associated with 21 threat actors. They extracted

features from IOCs, TTPs, and malware and optimized the model to achieve threat actor attribution accuracy of 83.2% (associating a threat actor with a threat intelligence report).

Irshad et al. [64] used IOCs, TTPs, malware, and data about targeted countries, industry sectors, and targeted applications for feature extraction and training ML models (RF and SVM). Both algorithms achieved 84% accuracy in identifying the correct threat actor behind an attack. Similarly, Chen et al. [44] conducted a study on threat actor clustering and noted the positive outcomes of attribute analysis of attack indicators such as TTPs and IOCs using a decision tree (DT). To improve their results, the authors used natural language processing (NLP) to extract behavioral characteristics from unstructured cyberattack and threat reports. Irshad et al. [65] also explored a hybrid feature set comprising both technical features and behavioral features representing the attack context. The best-performing ML model (long short-term memory—LSTM) achieved prediction accuracy of 97%.

Saha et al. [66] have developed a two-level ML model for automatic APT attribution that first identifies malicious samples in the context of a threat campaign, and then identifies samples that were operated by the same threat actor. The model uses input from a range of heterogeneous files (both executables and documents) and searches for connections between features. It was trained on an MITRE reference dataset and a dataset of APT samples belonging to a number of APT threat groups. Somewhat similarly, Xiang et al. [57] approached threat attribution by merging a large real-world network behavior dataset with threat intelligence data and applying a k-means clustering algorithm. The proposed system (IPAttributor) assesses similarities and links between IP data and applies dynamic weighted threat segmentation to identify attacker entities. The highest reported output accuracy was 88.89%.

In addition to the methodologies and algorithms used in the studies above, other pattern or characteristic matching techniques considered in current research include the use of neural networks and DL algorithms. The effectiveness of the chosen technique varies depending on the number of attributes of each entry, the structure of the data, and the size of the datasets [68]. Incident report data were used to extract attack or TTP information in studies that deployed advanced DL methods. For example, Duan et al. [69] used textual information gathered from CTI reports and applied a multi-level feature fusion approach and a combination of classifiers to identify attack patterns. Ge et al. [70] addressed the real-life challenge of identifying threat actors as early as possible during an attack. They proposed a state-space model called ThreatMAMBA; the input data (IOCs, TTPs, and temporal relationships) were extracted from CTI information (MITRE ATT&CK) and encoded in a knowledge graph.

In a number of studies, DL or hybrid methods were applied to create classifiers that extracted threat actor behavioral information from operational contexts. For example, Böge et al. [71] used a hybrid BERT-based model to identify cyberthreat actors based on the comparison of command sequences used in the attack once the target is accessed; they used data collected by honeypot servers located in China, Europe, and the United States and identified 34 actors, predominantly using the Cobalt Strike framework. Basnet et al. [72] applied deep reinforcement learning to attribute APTs using dynamic behavioral data gathered from malware samples that were executed and analyzed within the Cuckoo Sandbox environment (capturing API calls, system changes, and network traffic). Beltrán et al. [73] proposed a cyber deception framework (CYDEC) to profile threat actors; they used behavioral signal data gathered from adaptive decoys and dynamic network movements deployed to engage attackers, capturing packet- and flow-level traces that are processed into low-level features for semantic inference.

Furthermore, Zhang et al. [74] use an LLM to construct a schematic representation of an attack's technique and model the attack implementation details with a higher layer

of granularity than that of the features extracted from technical cyberthreat descriptors. The proposed automatic system for threat attribution (APTChaser) exhibited a high mean reciprocal ranking.

1.3. Research Aim and Approach

As shown by the review of the extant research, current approaches to threat actor attribution are still characterized by high uncertainty about the correctness of the identification of the threat actor group behind a specific cyberattack. This is due in part to the lack of reliable technical input to assist the time-consuming threat actor identification process [75,76]. The main aim of this study is to explore the feasibility of creating an effective ML model for threat actor attribution using cybersecurity intelligence. The choice of input features and the feature extraction method may significantly influence the performance of ML models. The study experiments with the use of an LLM model to extract information from cyberthreat reports and provide input to a model training dataset. The dataset uses threat actor TTPs as well as cyber incident context data as input features.

The study utilizes the MITRE ATT&CK Framework [17] classification, which comprises a total of 203 techniques and 453 sub-techniques, with identified threat actor groups deploying specific subsets of these techniques and sub-techniques. We propose and test a novel framework for threat actor attribution based on the TTPs of the cyberattack and on the contextual characteristics of the cyberattack targets. The proposed feature set supports successful threat actor attribution without the need to collect data about the indicators at the lower levels of the Pyramid of Pain, such as IOCs, domain information, IP addresses, and hashes. This may increase the efficiency of the SOC team operations. Furthermore, this research contributes a prompt engineering method that allows the use of ChatGPT for extracting cyber incident characteristics and matching them to known TTPs.

The rest of the paper is organized as follows. The literature review in Section 1.2 explores cyberattacks, threat intelligence, and the current threat actor attribution landscape. Section 2 describes the data sources used in this study, including the ChatGPT-based data extraction and the experiment setup. Sections 3 and 4 present and analyze the findings of the experiments.

2. Materials and Methods

2.1. ML Algorithms and Data Source

Both ML and DL methods and implementation platforms are used across cybersecurity domains, including cyberattack detection [77]. In their extensive review of ML, DL, and AI methods in cybersecurity, Ozkan-Okay et al. [78] point out that ML models are still widely used in cybersecurity research. In this study, we chose to use ML rather than DL methods because they work well with relatively small, structured datasets compared with DL models [79] and are more transparent and better suited to sparse, high-dimensional binary data [80]. Based on their use in previous threat actor attribution studies [44,62,64,81], five traditional ML algorithms were selected for optimization testing: DT, RF, SVM, NB and KNN.

To build the ML models, it was necessary to first create a suitable dataset of cyber incidents including the threat actors conducting cyberattacks, the MITRE ATT&CK TTPs as observed during the attack, and the victim's geographical location and industry sector. Section 1.2 identifies several databases that could be utilized for this research; based on the desired fields available in the database, the number of records included, the accuracy of its data, and the frequency of updates to its data, the EuRepoC database [81] was selected to build the cyber incident dataset used in this research.

The EuRepoC was set up by the EuRepoC consortium (established in 2022); the aim was to support cybersecurity-related discussions and policy-making by providing a reliable cyber incident database and analytical frameworks. Trustworthy contributors such as European Union (EU) government agencies, including the German Federal Foreign Office and the Danish Foreign Ministry, helped build the database, which is available for public use and research [82].

The data fed to the repository are collected from publicly available sources; these are mostly high-level incidents that may also have a political impact; this may filter out some otherwise significant cybersecurity incidents, e.g., cyber espionage. In addition, the repository collection tends to focus on the larger European countries, which may lead to missing information on emerging cybersecurity actors [83]. Despite the potential for introducing bias, the dataset is used widely in cybersecurity research because of its reliability and availability [84,85]. We chose to use the repository for the same reasons; as the study is an exploratory one, we did not include methods for missing data mitigation in the scope of the research; a discourse on the issue can be found in [84].

This research used the 4 March 2025 downloadable version of this database (eurepoc_global_dataset_1_3.xlsx) [81]. These data consist of 3414 rows and 85 columns. Each row contains data related to a specific cyber incident, including the description of the cyberattack, the date when the attack started and the date the attack ended, the date the attack was added to the database, a description of the attacker (e.g., the name and the country of the threat actor and the threat group), a description of the victim (e.g., industry sector and country name), and the cyber incident report or article source URLs. The database record also includes information about the response to the attack and its impact, the source of the attribution, and the number of threat actor group attributions for each cyber incident.

The following fields were used to create this study's dataset of cyber incidents:

1. incident_id

   The *incident_id* field indicates the specific cyberattack event that occurred.

2. receiver_country

   The *receiver_country* field specifies the victim's geographic location.

3. receiver_category_subcode

   The *receiver_category_subcode* field specifies the victim's industry sector.

4. initiator_name

The *initiator_name* field indicates the threat actor group name. This attribute is an essential data point that is used to test and validate the threat actor attribution model proposed in this study. This study focuses on the 'advanced persistent threat' (APT) actor groups (group name prefix APT). The APTs were selected as APT attacks are among the most dangerous ones in cyberspace. Often, they are hard to detect as the highly skilled malicious actors apply a range of tactics and use different tactics to carry out covert and sophisticated attack campaigns [86].

5. number_attributions

The *number_attributions* field shows how many threat actor groups were attributed to the cyber incident. As the attribution was based on information sourced from different channels (e.g., self-attributions, technical reports prepared by IT companies, political, direct, or anonymous statements in the media, legal action proceedings), some incidents were attributed to multiple threat actors. In this study, we limited the scope to cybersecurity incidents that had been attributed to one threat actor group only (number_attributions = 1) to avoid ambiguity.

6. sources_url

The *sources_url* field includes one or more cyber incident links to articles or report links that contain additional information about the specific cyberattack. This field was selected as the study used the articles to extract the TTPs associated with the specific cyber incidents.

2.2. LLM-Based TTP Extraction Prompt Engineering

In their systematic literature review, Zhan et al. [29] point out that LLMs are particularly suitable for processing threat intelligence text; the authors identify CTI information extraction as a key application area. A decoder-only transformer architecture such as the GPT model series is well-suited to the analysis of security reports [87]. Ozkan-Okay et al. [78] also note that ChatGPT is already used extensively for acquiring intelligence information. This study used an AI-supported TTP data extraction method. In their study of the effectiveness of AI in sentence parsing, Sewunetie et al. [88] found that ChatGPT version 3.5 was capable of understanding, parsing, and extracting requested information from input prompts. We used ChatGPT version 4.0 to extract specific TTPs from a reference cyber incident dataset that contained cyber incident report URLs in the field *sources_url*.

As reported in [89], prompt engineering is commonly used as a means of adapting LLMs to security tasks. In this study, the ChatGPT prompt. was engineered specifically to achieve high response accuracy. As shown below, the prompt was very specific and detailed. It instructs ChatGPT to extract TTPs from the sources associated with the specific cyber incident, using the TTP definitions in the MITRE ATT&CK Framework. As recommended in [90], we applied a grounding approach to minimize the occurrence of hallucinations in the ChatGPT response: the prompt provided step-by-step reasoning instructions and clear output formatting rules. Table 1 shows an example of how the prompt was applied to a specific report [91].

Table 1. Example: The ChatGPT prompt engineered to extract MITRE ATT&CK Framework TTP information from a specific report.

Prompt Component	Description
Steps to follow	Step 1. Visit the report link and read the information presented. Step 2. Identify specific MITRE ATT&CK Framework techniques and sub-techniques found in the body of the report that are explicitly stated or very strongly inferred by the text. Vague inference is not acceptable. Step 3. Extracted techniques and sub-techniques must be completely based on the information found in the report and not based on any external knowledge of the threat actor or the attack. Step 4. Each extracted technique and sub-technique should be supported with evidence in the format of a direct quote from the report for user validation. If the article is not in English, then translate the direct quote to English and do not output the quote in the original language. Only include the direct quote in the Supporting Evidence field in the output format template. Step 5. All techniques and sub-techniques must be based on the information found on https://attack.mitre.org/techniques/enterprise/ (accessed on 20 September 2025) and should be able to be justified based on the direct quote and relating to the technique or sub-technique description found on https://attack.mitre.org/techniques/enterprise/ (accessed on 20 September 2025). Step 6. Avoid unnecessary enumeration of sub-techniques if they are inferred and not explicitly mentioned. Step 7. Follow steps 1–6 again and verify that the results are the same. If they are not, then repeat 1–6 again until the results are consistent.

Table 1. *Cont.*

Prompt Component	Description

Output Format	The output must be in the following JSON format:
	{
	“incident-techniques”: [
	“TXXXX”,//Technique or Sub-technique Name—Supporting Evidence
	“TYYYY”//Another Technique or Sub-technique Name—Supporting Evidence
	],
	‘targeted-countries’: [
	‘Country1’,
	‘Country2’
	],
	‘targeted-industries’: [
	‘Industry1’,
	‘Industry2’
	],
	‘URL’: ‘Report URL’
	}

Additional clauses	Do not include any MITRE ATT&CK Framework techniques or sub-techniques unless they are directly supported by a quote from the report. Do not infer techniques based on the known behavior of a threat actor or group, unless these techniques or sub-techniques are strongly inferred within the report. Each technique or sub-technique must be directly justified by a quote from the report, which must be included word-for-word in the JSON output. If a technique cannot be explicitly tied to a quote, it must be excluded. Do not list techniques based on general attack stages or assumptions; only include those with specific textual evidence. If any technique is included without a matching direct quote from the article, the output is invalid. If the report explicitly lists any MITRE ATT&CK techniques or sub-techniques using MITRE technique IDs (e.g., T1059 or T1566.001), all of these must be extracted into the JSON output. Do not omit any technique listed with a T-ID, even if it is not described in full descriptive text. If a MITRE ATT&CK matrix or table is included in the report, all techniques in that matrix must be extracted and included in the output. Any omission of an explicitly listed technique or sub-technique invalidates the output. Strongly inferred techniques and sub-techniques must be included if they are supported by the report text. Inference is allowed only if it is strongly supported by specific quotes or sentences in the report and can be directly mapped to a MITRE ATT&CK technique. Vague, weak, or assumed inferences must be excluded. Provide the supporting evidence after the JSON output, in addition to it, for manual verification of the techniques. Before responding to this request, ensure that all these instructions, clauses, and formats are strictly followed.

As noted in [92], results may start to lose their accuracy over time with continued use of the service, due to errors and inconsistencies caused by ChatGPT's learning and adaptation to its own responses. To mitigate this risk and ensure that ChatGPT strictly followed the instructions as presented in the crafted prompt, a 'restart' prompt was sent between requests to ensure that these instructions were followed explicitly: 'Forget everything you have learnt, clear memory, and start a new instance'.

The EURepoC database was scanned by the authors to identify duplicate reports and reports that did not contain TTP information. Overall, ChatGPT extracted TTPs from 783 reports out of 800 preselected reports. Each report represented a cybersecurity incident.

The authors conducted a subjective expert verification of the output: they manually analyzed the supporting evidence (the direct quote provided by ChatGPT) for every technique or sub-technique identified by ChatGPT to ensure that the source report was accurately associated with the technique or sub-technique and that the direct quote clearly supported the ChatGPT response. To do this, the first author (who has substantial industry experience in cybersecurity) identified the source of the direct quote, located the direct quote and checked its content in the context of the incident described in the report, and verified the match between the incident and technique or sub-technique identified by ChatGPT. If a ChatGPT response was found invalid, it was removed from the results. To provide a measure of consistency, the second author checked the verification of every 10th incident. All checked verifications were found accurate. This process does not constitute a full quantitative validation of the TTP extraction quality and does not provide the data for calculating quantitative metrics. In particular, calculating metrics such as recall and F1-score would require an independently created verified 'true' TTP content annotation of each report, which was not available.

If a ChatGPT response was judged ambiguous, the following 'repeat' prompt was given to ChatGPT, requesting the model to provide more information: 'Reevaluate each technique and sub-technique and compare it to the MITRE ATT&CK Framework descriptions and remove any that do not adhere to the provided instructions and criteria'. The aim of this prompt was to make ChatGPT analyze its findings and provide additional justifications or evidence to support techniques it identified and remove techniques that were weakly inferred.

A total of 352 unique MITRE ATT&CK techniques and sub-techniques were extracted. As shown in Figure 1, there was a wide variance in the number of TTPs that ChatGPT was able to extract from each usable report, ranging from one TTP to 35 TTPs extracted from a report.

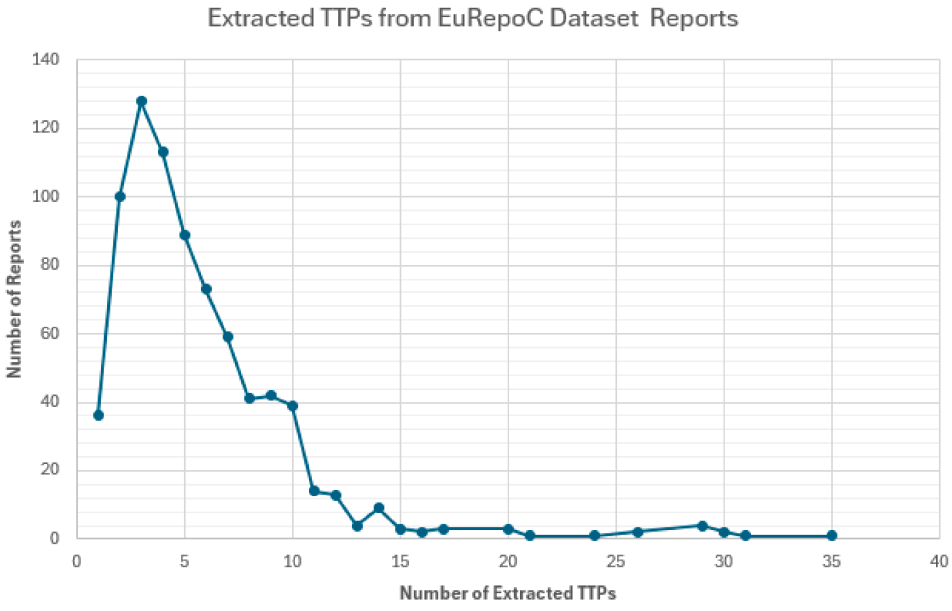

Figure 1. Number of extracted TTPs vs. number of reports. The x-axis represents the number of extracted TTPs from a single cyber incident report, and the y-axis represents the total number of reports that produced each number of TTPs.

2.3. Cyber Incident Dataset

To create a cyber incident dataset for training and testing the ML models, we selected APT actors who met the following criterion: to be associated with at least 15 cyber incidents. The selection process resulted in a dataset A of ten MITRE threat actors (Table 2).

Table 2. Threat actors and the number of attributed incidents in the study dataset.

Threat Actor ID	Number of Attributed Incidents
APT_32	18
APT_27	32
APT_37	46
APT_34	56
APT_41	58
APT_45	82
APT_35	87
APT_28	112
APT_44	123
APT_29	169

The resulting database of analyzed cyber incidents comprised 783 incidents, each attributed to one of the threat actors listed in Table 2. Each record of the dataset contained the threat actor ID and the extracted TTP, the targeted country/region, and the targeted industry sector/subsector. Table 3 describes the variables included in the database.

Table 3. Cyber incident dataset variables.

Cybersecurity Incident Variables	Number of Unique Values
APT threat actor	10
TTPs	352
Targeted countries/regions	84
Targeted industry sectors/subsectors	35

To construct the study dataset, we created a feature set that included all unique TTP, country/region, and industry sector values. The decision to include all unique values of the contextual variables (targeted country/region and targeted industry sector/subsector) was informed by prior work where results indicated that contextual features improved the accuracy of threat actor attribution [61,63,65]. The extracted TTPs were distributed unevenly; we decided to include all unique TTP values to avoid information loss and to capture rare but potentially important signals. We applied one-hot encoding [92] to the variables described in Table 3, mapping every variable to a binary value of 0 or 1. The resulting study dataset T contained 783 data rows representing individual attributed cyber incidents. Each row was labeled with a threat actor ID from the dataset A and comprised 471 feature columns of binary values. The resulting structured cyber incident dataset T was slightly imbalanced, as 52% of the data points (the cyber incidents) belonged to three classes only (APT_29, APT_44, and APT_28).

2.4. Experiment Design

Five ML algorithms were selected and used to build an ML multiclass classification algorithm. Four of them (DT, RF, SVM, NB) were selected based on their use in previously published research [44,62,64,93] (refer to Section 1.2). The fifth ML algorithm (k-nearest neighbors—KNN) is a supervised learning algorithm that classifies a new data point based on the classification of similar data points in the training dataset [94]. Dolai et al. [95] proposed a framework for the classification of cyber threat reports based on their APT groups. The authors compared the performance of five ML algorithms: XGBoost, AdaBoost, SVM, deep neural network (DNN), and KNN; KNN produced the best classification results (93.17% accuracy). Based on the above, KNN was considered suitable for this research as threat actors are identified based on the TTPs observed during an attack and the similarities between targeted countries/regions and industry sectors.

To deploy and experiment with the ML algorithms, a series of Python 3.12 scripts was created by utilizing the Python scikit-learn open-source ML library in a Python IDE (PyCharm Community Edition 2023.3 [96]). The scripts were written and tested on an Asus ROG Strix G16 laptop with 64 GB of RAM, a 13th Gen Intel i7 CPU, and an Nvidia GeForce RTX 4060 GPU, running on Windows 11 Home Edition. The dataset and the Python code are available as Supplementary Materials.

In all experiments, the input dataset was split into a training dataset (70%) and a testing dataset (30%). To ensure consistency across all algorithms regarding splitting the training and testing data, the 'random state' value was set to 7 across all models. To reduce overfitting and bias due to the majority of TTPs extracted from a relatively small number of cybersecurity reports, a stratified 10-fold cross-validation method [97] was adopted in all experiments where the input dataset included all TTPs features (refer to Sections 3.1 and 3.2). Stratified 3-fold cross-validation was used in the experiments that compared model performance across specified TTP feature ranges (refer to Section 3.3).

The performance of the models was evaluated using the performance metrics of precision (PR), recall (RC), F1-score (the harmonic mean of PR and RC), and mean model accuracy (mean AC) and standard deviation (STD).

While the mean AC is an effective indicator of the overall model performance, the F1-score is also a significant performance metric in the context of the experiment: both false positives (the threat actor attributed to the incident is not the one behind the incident) and false negatives (the threat actor behind the incident was not recognized) have a significant impact on the operational and managerial decisions made based on the classification output. To account for the unbalanced class size, the weighted values of PR, RC, and F1-score were considered.

3. Experiments and Results

In the three subsections following, we provide the results of six experiments, each series involving a different feature selection method. To illustrate the step-wise approach in training and testing the different ML models, we present performance data per class for KNN models, and average performance data for all models. The KNN models were tested with three different values of K (1, 2, and 3); in the results below, K was set to 1, as the best-performing parameter.

3.1. Threat Actor Attribution Using Unweighted Features

In the first two series of experiments, the ML models were trained and tested using all encoded TTP features in the cyber incident dataset T. In the second series, all encoded features of the TTPs and all encoded cyberattack target features (i.e., the data about the targeted country/region, and industry sector and subsector of the attack receiver) were included as input features in the training and testing data sets. Table 4 shows the per-class metrics of the KNN model, while Table 5 shows the weighted PR, RC, and F1-score values for the multiclass classification, and the mean AC and STD achieved by the five models (applying 10-fold cross-validation). As seen, all models achieved both higher accuracy and higher F1-score when using all features.

Table 4. KNN model performance (per class) using unweighted features.

Actor Class	PR		RC		F1-Score	
	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features
APT_27	0.62	0.60	0.50	0.60	0.56	0.60
APT_28	0.46	0.63	0.60	0.87	0.52	0.73
APT_29	0.74	0.83	0.81	0.92	0.77	0.87
APT_32	0.50	0.10	0.33	0.33	0.40	0.50
APT_34	0.53	0.85	0.47	0.65	0.50	0.73
APT_35	0.66	0.77	0.79	0.10	0.72	0.87
APT_37	0.40	0.10	0.27	0.73	0.32	0.85
APT_41	0.32	0.75	0.60	0.60	0.41	0.67
APT_44	0.74	0.87	0.72	0.93	0.73	0.90
APT_45	0.75	0.87	0.32	0.46	0.45	0.60

Table 5. ML model performance metrics using unweighted features (in percentage).

ML Algorithm	Weighted PR		Weighted RC		Weighted F1-Score		Mean AC		STD	
							10-Fold Cross-Validation			
	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features
KNN	63.00	81.00	61.00	80.00	61.00	79.00	55.29	79.94	4.77	4.71
DT	60.00	85.00	59.00	83.00	59.00	83.00	55.68	84.92	4.78	3.14
RF	72.00	91.00	69.00	90.00	69.99	90.00	65.75	91.82	5.77	2.94
SVM	61.00	84.00	59.00	83.00	59.00	84.79	55.91	84.79	7.76	5.11
NB	55.00	77.00	55.00	75.00	51.00	75.00	55.30	69.86	5.29	5.68

3.2. Threat Actor Attribution Using Weighted Features

In the subsequent experiments, several different approaches were applied to improve the models' performance by applying weights to the selected features.

3.2.1. Rarity-Based Feature Weights

In this approach, the encoded TTP and attack target features were assigned weights based on their usage frequency in the cyber incident dataset T. The assumption underlying the approach was that the more uncommon a TTP or a characteristic of the targeted cyberattack victim was, the higher the weight it should be given when used for threat actor attribution. Table 6 shows the algorithm for calculating rarity-based feature weights.

Table 6. Assigning rarity-based weightings.

Step	Description
Input and Output 1	Input: Dataset D containing one-hot encoded features for threat activity; target variable y = APT identifier (apt_num). Output: Weighted and standardized feature matrix X′ ready for KNN classification.
Load dataset	Read the file 'One-Hot_Encoded_Threat_Data.csv' into the data frame *D*.
Feature group extraction	(a) *Identify* columns in *D* beginning with the following prefixes: - 'ttps_' → TTP-related features - '*receiver_country_*' → *country-related* features - '*receiver_category_*' → sector-related features - '*receiver_subcategory_*' → *industry-related* features (b) Combine all identified columns to form the complete feature set *F*.
Rarity-based weight calculation	(a) *For* each feature f in *F*, count the number of unique APT identifiers for which f = 1. *Denote* this count as *c_f* (feature usage *frequency).* *(b)* Determine *c_max*, the maximum of all usage counts. (c) Assign an amplified *rarity-based* weight *w_f* to each feature according to the formula: *w_f* = ((1/*c_f*)/(1/*c_max*))^3 if *c_f* > 0, else 0. (d) Apply this weighting procedure *separately* to each group of features (TTP, *country*, sector, and *industry).*

Table 6. *Cont.*

Step	Description
Rarity-based weight application	(a) Extract the feature matrix X and label vector y from D. *(b)* Standardize all features in X using Z-score normalization. (c) *For* each column f in X, multiply all standardized values *by* the corresponding weight w_f. *(d)* The resulting matrix X' represents the weighted and standardized feature set. Data

Similarly to above, two series of experiments were conducted, training and testing all ML models on an input dataset. In the first series, the input file included the TTP features in the cyber incident dataset T (weighted, based on rarity). In the second series, the input file included all features in the dataset T, rarity-based weighted.

Table 7 shows the per-class metrics of the KNN model, while Table 8 shows the weighted PR, RC, and F1-score values for the multiclass classification, and the mean AC and STD achieved by the five models (applying 10-fold cross-validation).

As in the previous tests, the mean accuracy of the models was significantly higher when all features were included. While the models' mean accuracy using weighted and unweighted features was comparable for both types of feature selection (TTP features only, and all features, respectively), the respective weighted F1-scores of the models using all weighted features were higher than the weighted F1-scores of the models using all unweighted features.

Table 7. KNN model performance (per class) using rarity-based feature weightings.

Actor Class	PR		RC		F1-Score	
	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features
APT_27	0.71	0.86	0.50	0.60	0.59	0.71
APT_28	0.49	0.66	0.60	0.83	0.54	0.74
APT_29	0.79	0.92	0.79	0.90	0.79	0.91
APT_32	1.00	0.67	0.33	0.33	0.50	0.44
APT_34	0.56	0.82	0.53	0.82	0.55	0.82
APT_35	0.64	0.83	0.75	1.00	0.69	0.91
APT_37	0.36	0.88	0.27	0.93	0.31	0.90
APT_41	0.30	0.88	0.70	0.70	0.42	0.78
APT_44	0.77	0.86	0.79	0.84	0.78	0.85
APT_45	0.87	0.88	0.46	0.75	0.60	0.81

Table 8. ML model performance metrics using rarity-based feature weightings (in percentage).

ML Algorithm	Weighted PR		Weighted RC		Weighted F1-Score		Mean AC		STD	
							10-Fold Cross-Validation			
	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features	TTP Features Only	All Features
KNN	68.00	64.00	64.00	83.00	64.00	83.00	55.16%	3.86	79.30	3.96
DT	60.00	85.00	59.00	83.00	59.00	83.00	55.68%	4.78	84.92	3.14
RF	72.00	91.00	69.00	90.00	69.00	90.00	65.75%	5.77	91.82	2.94
SVM	64.00	84.00	60.00	83.00	81.00	83.00	54.77%	6.38	82.87	3.98
NB	49.00	78.00	44.00	77.00	43.00	77.00	38.80%	7.31	73.81	5.55

3.2.2. Pre-Set Feature Weights

This manual weighting approach explored the impact of a number of combinations of predetermined weights in the range of 1 to 3 for the different feature categories. It was found through experimentation and testing that the highest accuracy levels were achieved when TTP features were given a weight of 1, and the attack target features were given a weight of 3. Table 9 shows the algorithm.

Table 9. Assigning flat weights.

Step	Description
Input and Output	As in Table 6
Load dataset	As in Table 6.
Feature group extraction	As in Table 6
Feature matrix standardization	(a) Apply z-score normalization to all columns in X using a standard scaler: *X_scaled = Standardize(X)* (b) Ensure each feature has zero mean and unit variance before weighting.
Apply manual flat weights	(a) Define weighting rules: *- TTP-related features: weight* = 1 *- Contextual features (country, sector, industry)*: *weight* = 3 (b) For each feature f in F, apply the corresponding multiplier to the standardized matrix: If $f \in$ *contextual features* $\rightarrow$ *multiply column X_scaled[:, f]* $\times = 3$ *Else* $\rightarrow$ *multiply column X_scaled[:, f]* $\times = 1$ (c) Return the resulting manually weighted feature matrix X'.

Table 10 shows the per-class metrics of the KNN model, while Table 11 shows the weighted PR, RC, and F1-score values for the multiclass classification, and the mean AC and STD achieved by the five models (applying 10-fold cross-validation). Overall, these results show an improvement in comparison to the previous sets of results. The average F1-score was improved.

Table 10. KNN model performance (per class) using manually determined flat weighting for all features.

Class	PR	RC	F1-Score
APT_27	0.78	0.70	0.74
APT_28	0.73	0.90	0.81
APT_29	0.89	0.98	0.94
APT_32	1.00	0.33	0.50
APT_34	1.00	0.88	0.94
APT_35	0.96	1.00	0.98
APT_37	0.88	1.00	0.94
APT_41	0.67	0.60	0.63
APT_44	0.95	0.88	0.92
APT_45	0.92	0.79	0.85

Table 11. ML model performance metrics using manually determined flat weighting for all features.

ML Algorithm	Weighted PR	Weighted RC	Weighted F1-Score	Mean AC	STD
				10-Fold Cross-Validation	
KNN	89.00	88.00	88.00	91.95	2.58
DT	85.00	83.00	83.00	84.92	3.14
RF	91.00	90.00	90.00	91.82	2.94
SVM	91.00	90.00	90.00	90.80	3.30
NB	80.00	78.00	78.00	76.12	4.67

3.2.3. Hybrid Feature Weights

In this last series of experiments, the TTP features were assigned rarity-based weights (refer to Section 3.2.1), while the attack target features were assigned fixed weights when the weights of the attack target features were set to 3 (as this value gave the best accuracy).

Table 12 shows the per-class metrics of the KNN model, while Table 13 shows the weighted PR, RC, and F1-score values for the multiclass classification, and the mean AC and STD achieved by the five models (applying 10-fold cross-validation). Both the mean accuracy across the models and the averaged F1-score were improved for the KNN, SVM, and NB models.

Table 12. KNN model performance using rarity-based weightings for TTP features and predetermined flat weighting for the attack target features.

Class	PR	RC	F1-Score
APT_27	0.67	0.80	0.73
APT_28	0.87	0.90	0.89
APT_29	0.94	0.96	0.95
APT_32	0.67	0.33	0.44
APT_34	1.00	0.94	0.97
APT_35	0.96	1.00	0.98
APT_37	1.00	1.00	1.00
APT_41	0.73	0.80	0.76
APT_44	0.98	0.93	0.95
APT_45	0.82	0.72	0.82

Table 13. ML model performance metrics using rarity-based weightings for TTP features and predetermined flat weighting for the attack target features.

ML Algorithm	Weighted PR	Weighted RC	Weighted F1-Score	Mean AC	STD
				10-Fold Cross-Validation	
KNN	91.00	91.00	90.00	92.59	2.68
DT	84.00	83.00	83.00	84.92	3.14
RF	91.00	90.00	90.00	91.82	2.94
SVM	94.00	94.00	94.00	92.20	2.60
NB	83.00	82.00	82.00	78.80	4.48

To check the accuracy of the results, the final version of the KNN model was modified to include an interactive data input function in the Python script. The function prompts the user to input TTPs, target country, and target industry information using the dataset T format. Once the user enters this information, the model performs the threat actor attribution classification and outputs the predicted threat actor attribution.

Using the input function, the model was tested on new samples of cyber incident data (five for each threat actor class). As shown in Table 14, the average classification accuracy achieved was 86.00%.

Table 14. Manual testing of the KNN model.

Threat Actor ID	Classification Accuracy
APT_27	0.80
APT_28	0.80
APT_29	1.00
APT_32	0.40
APT_34	1.00
APT_35	1.00
APT_37	0.80
APT_41	0.80
APT_44	1.00
APT_45	1.00
Average	0.86

An additional experiment was conducted to explore the impact of small-sized classes on the model performance. When the smallest class APT_32 (with the lowest weighted F1-score of 0.44) was removed to create a more balanced input dataset, the model's mean AC increased slightly to 93.19%.

3.3. Statistical Testing

To test the statistical significance of the differences in the ML performance as reported above, Friedman tests were conducted on the per-fold accuracy scores from the stratified 10-fold cross-validation for each feature-weighting configuration (the fold treated as the repeated-measures unit). The results (Appendix A) indicated a statistically significant difference among the five algorithms ($p < 0.001$ in all configurations).

In the best-performing (hybrid-weighted) configuration, the post hoc Wilcoxon signed-rank tests with Bonferroni correction ($\alpha = 0.005$ for 10 pairwise comparisons) showed that KNN (92.59%), SVM (92.20%), and RF (91.82%) each significantly outperformed DT (84.92%) and NB (78.80%, $p < 0.002$), but were not significantly different from one another.

3.4. Threat Actor Attribution by TTP Ranges

To explore the impact of the high number of TTP features in the feature set of the cyber incident dataset T, we split the dataset T into subsets based on the number of TTP features with a value of 1 (i.e., the number of TTPs extracted from the respective cyber incident report), and removed the TTP feature columns that had 0 values only. We aggregated the subsets to create input datasets with counts of TTPs in the ranges shown in the first column of Table 15. Three tests were performed, using the KNN algorithm to build the ML classifier, using: (i) unweighted TTP features only (as described in Section 3.1); (ii) rarity-based weighted TTPs features only (as described in Section 3.2.1); and (iii) rarity-based TTP features of and predetermined flat weighting of the attack target features (as described in Section 3.2.2). Table 15 shows the mean AC for each of the tested ranges, using stratified 3-fold cross-validation.

Table 15. Mean AC across TTP ranges using three KNN models.

TTP Count Range	Number of Samples in the Input Dataset	Mean AC (3-Fold Cross-Validation)		
		Unweighted TTP Features Only	Rarity-Based Weighted TTP Features Only	Rarity-Based Weighted TTP Features with Flat Weighted Attack Target Features
1–3	265	0.615	0.623	0.868
4–6	274	0.463	0.504	0.785
7–9	145	0.359	0.365	0.710
10–13	68	0.399	0.413	0.617
14–17	16	0.311	0.189	0.567
18+	15	0.533	0.600	0.533

4. Discussion

This research identified ML algorithms that have shown effectiveness in threat actor attribution. The effectiveness of the respective models was explored and compared using different methods of feature selection and weighting.

4.1. Results Overview

Overall, the NB models performed consistently worse than the other four ML algorithms. The highest mean AC was obtained by the KNN algorithm utilizing rarity-based weighted TTP features and predetermined weights for the attack target features. The second-best performing algorithm was SVM with a mean AC = 92.20% and weighted

F1-score = 94% (the highest across all models). The RF algorithm achieved the highest mean AC of 65.75% and the best F1-score of 69% when using TTP features alone. Using different types of feature weighting did not have an impact on the performance of the RF and DT models.

The KNN model responded most positively to using predetermined weighting, achieving the highest mean AC of 91.95% (with the attack target features weighted three times more than the TTP features). However, the weighted F1-score of the KNN model was relatively low (88%). Lastly, when TTP features were weighted based on rarity and the attack target features were given a flat weighting, the best-performing model (KNN) achieved a mean AC of 92.59% and a weighted F1-score of 90% (second highest across all models). The manual verification of the results achieved by the KNN model using hybrid feature weighting indicated that the model was performing well for all classes. Additionally, the user input function provides a means for user testing of the black-box ML model, thus increasing user confidence in the predicted threat actor attribution.

4.2. ML Models and Feature Selection Methods for Threat Actor Attribution

Returning to the research question guiding the study, it may be concluded that four of the explored ML models (DT, RF, SVM, and KNN) can be used to build effective ML models. Regarding the use of TTP features only in the input datasets, the experiments indicated that using the extended feature space including the TTP features, which represent the cyberattack methods and the features characterizing the cyberattack context (geographical location and industry), significantly improved all models' performance.

In particular, when using TTP features alone, the classification accuracy of the four models varied between 55.16% and 65.75%; the best-performing model (RF) achieved a mean AC of 65.75% for both weighted and unweighted TTP features. When the attack target features were added, the classification accuracy varied between 79.30% and 92.59%.

The choice of feature weighting method had a significant impact on the performance of the KNN and SVM models only (with the classification accuracy increased from 70.94% and 84.78% to 92.59% and 92.20% for the KNN and SVM models, respectively); the classification accuracy of the DT and RF models was not affected by applying feature weighting (highest AC achieved was 84.92% and 91.82%, respectively).

The highest mean AC values were achieved by KNN and SVM, respectively (applying rarity-based weighting to the TTP features and flat weighting to the attack target features); the highest F1-scores were also achieved by the same model/feature selection combination but in reverse order (the F1-score of SVM was 94.00% compared with the F1-score of KNN, which was 90.00%). These results also indicated that setting the K value to 1 for the KNN algorithm did not cause significant overfitting. This is important as real-life cyber data includes information about well-known threat actors and about new entrants to the cybercrime scene.

4.3. Cyber Incident Classes and Feature Set Size

4.3.1. Class Sample Size

In this study, the dataset was constructed to include threat actors with more than 15 reported cyber incidents in the MITRE database; this resulted in five 'large' classes (sample sizes between 18 and 58) and five 'small' classes (sample sizes between 82 and 169). This affected the performance of the ML models. For example, the performance metrics of the class that had a number of samples very close to the cutoff point (APT_32, with 18 samples) were consistently the lowest, compared with the other classes.

When the KNN model was tested on new data, the classification accuracy of four of the smaller classes was below the average while the classification accuracy of four of the

large classes was above the average, reflecting imbalance. However, running the KNN model over a resampled dataset that excluded APT_32 showed only a small increase in the classification accuracy (0.65%).

4.3.2. Feature Set Size

As pointed out earlier, the cyber incident dataset T had a very large feature set as it included all TTPs extracted from the 783 cyber incident reports. Comparing the results from the tests described in Section 3.3, it can be seen that the actor attribution accuracy was considerably higher for datasets with a smaller number of TTPs (Figure 2). However, the results are inconclusive as the datasets with a higher number of TTP feature counts contained a small number of samples.

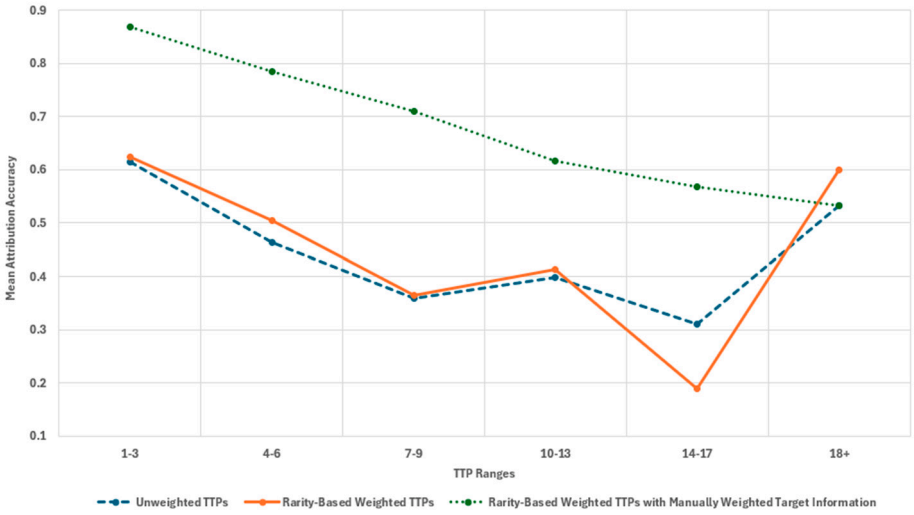

Figure 2. Mean attribution accuracy levels for ranges of TTP features. Note. The x-axis represents ranges of numbers of TTP features corresponding to TTPs extracted from cyber incident reports. The y-axis represents the threat actor attribution mean accuracy levels using KNN with three different types of feature selection.

Further analysis of the performance of the KNN model across datasets with different numbers of TTPs indicates that the highest classification accuracy when using TTP features only was achieved when the number of TTP features was between 1 and 6 or above 18, with a model performance trend similar for unweighted and weighted TTP features. This may indicate that the TTP features in these datasets included a higher number of unique TTPs compared with the other datasets, which may have included a higher number of common TTPs. Furthermore, when all features were used and weighted (rarity-based for TTP features, flat weighting for attack target features), the model performance tended to drop with the increase in the number of TTP features in the set. Ultimately, three models performed with the same accuracy for datasets with 18 or more TTP features for all three methods of feature selection and weighting.

The number of TTP features per sample was determined by the ChatGPT output. With effective prompt engineering, providing adequate instructions detailing precisely how to extract the TTPs from each report and including strict criteria to be followed, accurate TTP extraction can be performed. However, mistakes can occur in this process, with ChatGPT often producing vague inferences. It was necessary to manually verify ChatGPT's responses to ensure the correctness of the threat extraction. This process may have affected the quality of the threat extraction.

4.4. Comparison with Related Work

Table 16 presents a comparison between this study and four related threat actor attribution studies that were reviewed in Section 1.2. The reported accuracy of the studies varies from 54.42% to 96%. The highest reported accuracy values for ML-based and DL-based approaches are comparable. This study also compares favorably to most of the studies in terms of achieved accuracy. While the fourth study (an ML-based study) has a higher accuracy, it uses more feature categories, making it harder to implement in real-life contexts.

Table 16. Comparison with other relevant work.

Study	IOC	Malware	TTP	Attack Target Data	ML Algorithms	Highest Threat Actor Attribution Accuracy
Duan et al. [61]	✓	✓	✓	✓	Heterogeneous graphs	54.42%
Kida et al. [62]		✓			RF, SVM, NB. K-means clustering	89%
Xiao et al. [63]	✓	✓	✓		Heterogeneous graphs	83.2%
Irshad & Basit Siddiqui, 2023 [64]	✓	✓	✓	✓	RF, DT, SVM	96.00%
Böge et al. [71]					BERT, Hybrid transformer + CNN	95.13%
Basnet et al., 2025 [72]		✓			Deep Reinforcement Learning	94.12%
Duan et al. [69]			✓		Multi-level feature fusion + Dempster–Shafer evidence theory	89.99%
Ge et al. [70]	✓		✓		Mamba (a taw-spaced neral network archirecture)	91.49%
This study			✓	✓	KNN, DT, RF, SVM, NB	93.19%

A broad comparison of attribution methods and evaluation metrics provided in [98] shows that attribution accuracies in empirical research involving the use of ML/DL models (including LLMs) have improved from around 58% in earlier work to 95–97% in more recent studies. While the accuracy achieved in this research is aligned with the trend exhibited in prior research, additional corroboration would still be needed if the evaluated model is used for real-life investigation and decision-making.

4.5. Effectiveness and Prediction Challenges

While SVM performs well in high-dimensional spaces [99] and KNN is easy to use, both models face challenges in terms of computational demands and prediction accuracy when data exhibit unknown patterns. As the dataset used in this research was relatively small, the experiments were not hampered by high computational resource demands such as processing time or insufficient memory. In general, the computational complexity of the KNN algorithm decreases with the increase of the number of data samples and the number of features describing the data; as shown in [100,101], it is possible to reduce the computational overhead of KNN models by pre-processing to decrease the dimensionality and by selecting a suitable search strategy. SVM models may also become too slow and require significant memory resources in the case of large datasets [102].

Second, both SVM and KNN models may misattribute incidents if new threat actors appear or if known actors change their attack approaches significantly. However, the algorithms can be extended to identify and notify such data points [103,104]. Once notified, the SOC team will need to conduct further investigations.

5. Conclusions

Lack of automation and reliance on manual threat intelligence hamper the effectiveness of SOC teams [105]. Saha et al. [67] note that the existing tools used by SOCs in cybersecurity incident investigations may lack the ability to derive accurate APT attribution due in part to the diversity and complexity of the information sources. The findings of their study indicate that cybersecurity experts and SOC teams need reliable TTP attribution to support cybersecurity incident investigation and incident responses and to support consequent strategic decision-making. While the majority of TTPs can be attributed to a relatively small number of countries [106], precise country attribution can inform long-term strategies and control [67,107].

From a theoretical perspective, the proposed approach contributes to building a process theory for CTI practice [108]. The outcomes of this study may help SOCs and cybersecurity professionals estimate the number and types of data points that are required to adequately attribute a cyberattack to a threat actor and how to improve the accuracy of the classification by weighting the input dataset features. To this end, this research makes several contributions. First, it demonstrates how specific ML algorithms can be used to model threat actor attribution with high accuracy, using TTP data as well as data about the attack context. Second, the study has demonstrated the effectiveness of using an LLM to analyze and extract TTPs from cyber incident reports, following provided TTP specifications. Third, the study has contributed a multidimensional dataset of 783 samples attributed to 10 threat actors that can be used in related research.

The study has several limitations. Only five traditional ML algorithms were used in the experiments; more advanced DL models were not explored. There were also a number of limitations related to the study dataset and the encoding approach: the dataset had high dimensionality, while the number of threat actors was relatively small, leading to sparsity and distance metric distortion in the KNN and SVM models; the number of TTP features with a binary value of 1 (i.e., the TTP features specific to the sample) varied widely, with 69% of the samples in the dataset represented by six or fewer TTPs; the class size also varied, with the smallest class represented by only 18 samples. Furthermore, while the proposed approach is time- and cost-efficient, the accuracy of the model is not sufficiently high to warrant remedial action or long-term planning without some additional supporting investigation. Furthermore, the study relied on expert evaluation of the LLM output, which may have introduced bias in the study dataset despite the risk mitigation approach that was applied.

Lastly, the threat actor attribution approach that was proposed and evaluated in this study is inherently static. Even though TTPs are positioned at the top of the Pyramid of Pain and are considered reliable and stable cyberattack attributes, threat actors may alter their techniques, and new threat actors may appear. To ensure trustworthy threat actor attribution, the most up-to-date information should be used for model training. However, this study used only one source to identify techniques and did not consider altered or newly observed existing techniques, or emerging techniques yet to be defined by the MITRE ATT&CK Foundation. As KNN and SVM are closed-set classifiers, they may misclassify data samples that do not match any of the known classes; special modifications are needed to enable the models to flag unknown data rather than classify them.

There are several aspects of this research that can be explored in future research to expand the capabilities of threat actor attribution. A more balanced dataset that contains more classes (i.e., more threat actors) and more cybersecurity incidents could be used in an ML or a DL model, which may increase the accuracy of the classification and achieve a better understanding of the effect of using hierarchically grouped TTPs. Future work could include a formal quantitative validation of the LLM-based TTP extraction process by

creating a benchmark set of cyber incident reports correctly annotated with the relevant TTPs. This would provide a more rigorous assessment of extraction reliability than the manual expert verification process used in this study. The manual testing conducted within this research can be expanded upon to explore variance in threat actor attribution by using this dataset for training the final KNN model against cyber incidents that were not included within the training dataset and additional real-world incidents, or against further subsets of TTPs from reports already utilized in the training dataset. Moreover, future research can explore the impact of the inclusion of additional cyber incident attributes from the Pyramid of Pain as a means of refining the classification model output, such as IP addresses, hashes, domains, malware, or tools. Another avenue for further research is to investigate the effectiveness of other ML algorithms and improve the classification accuracy by using an ensemble ML model.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/fi18080433/s1 (dataset and Python code).

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

Funding: This research received no external funding.

Data Availability Statement: We are currently considering publishing the dataset in a dataset repository.

Acknowledgments: During the study, the authors used ChatGPT version 4.0 for the purposes of generating some of the data used in the study. The process is described in detail in Section 2. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

TTP	Tactics, Techniques, and Procedures
APT	Advanced Persistent Threat
PE	Prompt Engineering
LLM	Large Language Model
DL	Deep Learning
ML	Machine Learning
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
RF	Random Forest
DT	Decision Tree
NB	Naïve Bayes
SOC	Security Operations Center
IOC	Indicator of Compromise
AC	Accuracy
RC	Recall
PR	Precision
GenAI	Generative AI
NLP	Natural Language Processing

Appendix A. Configuration Testing

Table A1. Hybrid (rarity TTP and flat context = 3), all features: best configuration. Friedman test: $\chi^2(4) = 33.65$, $p < 0.000001$. Mean accuracy by algorithm: KNN: 92.59%, SVM: 92.20%, RF: 91.82%, DT: 84.92%, NB: 78.80%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	5.500	0.34375	No	No
KNN	RF	12.000	0.46094	No	No
KNN	DT	0.000	0.00195	Yes	Yes
KNN	NB	0.000	0.00195	Yes	Yes
SVM	RF	18.000	0.65234	No	No
SVM	DT	0.000	0.00195	Yes	Yes
SVM	NB	0.000	0.00195	Yes	Yes
RF	DT	0.000	0.00195	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	0.000	0.00781	Yes	No

Table A2. Flat weighting (TTP = 1, context = 3, all features): second-best configuration. Friedman test: $\chi^2(4) = 31.82$, $p = 0.000002$. Mean accuracy by algorithm: KNN: 91.95%, SVM: 90.80%, RF: 91.82%, DT: 84.92%, NB: 76.12%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	5.000	0.07812	No	No
KNN	RF	22.000	0.97656	No	No
KNN	DT	0.000	0.00195	Yes	Yes
KNN	NB	0.000	0.00195	Yes	Yes
SVM	RF	13.000	0.30078	No	No
SVM	DT	2.000	0.00586	Yes	No
SVM	NB	0.000	0.00195	Yes	Yes
RF	DT	0.000	0.00195	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	0.000	0.00195	Yes	Yes

Table A3. Unweighted, all features. Friedman test: $\chi^2(4) = 36.97$, $p < 0.000001$. Mean accuracy by algorithm: KNN: 79.94%, SVM: 84.79%, RF: 91.82%, DT: 84.92%, NB: 69.86%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	0.000	0.00781	Yes	No
KNN	RF	0.000	0.00195	Yes	Yes
KNN	DT	0.000	0.00781	Yes	No
KNN	NB	0.000	0.00195	Yes	Yes
SVM	RF	0.000	0.00195	Yes	Yes
SVM	DT	21.500	0.93359	No	No
SVM	NB	0.000	0.00195	Yes	Yes
RF	DT	0.000	0.00195	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	0.000	0.00195	Yes	Yes

Table A4. Unweighted, TTP features only. Friedman test: $\chi^2(4) = 18.98$, $p = 0.000791$. Mean accuracy by algorithm: KNN: 55.29%, SVM: 55.91%, RF: 65.75%, DT: 55.68%, NB: 55.30%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	17.000	0.92188	No	No
KNN	RF	0.000	0.00195	Yes	Yes
KNN	DT	20.500	0.84375	No	No
KNN	NB	18.000	1.00000	No	No
SVM	RF	0.000	0.00195	Yes	Yes
SVM	DT	22.000	1.00000	No	No
SVM	NB	25.000	0.82422	No	No
RF	DT	1.000	0.00391	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	17.500	0.58594	No	No

Table A5. Rarity-weighted, TTP features only. Friedman test: $\chi^2(4) = 28.80$, $p = 0.000009$. Mean accuracy by algorithm: KNN: 55.16%, SVM: 54.77%, RF: 65.75%, DT: 55.68%, NB: 38.80%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	19.000	0.71094	No	No
KNN	RF	0.000	0.00195	Yes	Yes
KNN	DT	21.500	0.92188	No	No
KNN	NB	0.000	0.00391	Yes	Yes
SVM	RF	0.000	0.00195	Yes	Yes
SVM	DT	17.000	0.57031	No	No
SVM	NB	0.000	0.00195	Yes	Yes
RF	DT	1.000	0.00391	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	1.000	0.00391	Yes	Yes

Table A6. Rarity-weighted, all features. Friedman test: $\chi^2(4) = 34.12$, $p < 0.000001$. Mean accuracy by algorithm: KNN: 79.30%, SVM: 82.87%, RF: 91.82%, DT: 84.92%, NB: 73.81%.

Algorithm 1	Algorithm 2	W	*p*-Value	Sig. $p < 0.05$	Sig. Bonferroni ($\alpha = 0.005$)
KNN	SVM	0.000	0.00391	Yes	Yes
KNN	RF	0.000	0.00195	Yes	Yes
KNN	DT	1.000	0.00391	Yes	Yes
KNN	NB	3.500	0.01172	Yes	No
SVM	RF	0.000	0.00391	Yes	Yes
SVM	DT	10.500	0.17578	No	No
SVM	NB	0.000	0.00391	Yes	Yes
RF	DT	0.000	0.00195	Yes	Yes
RF	NB	0.000	0.00195	Yes	Yes
DT	NB	0.000	0.00195	Yes	Yes

References

1. Zang, T.; Xiao, Y.; Liu, Y.; Wang, S.; Wang, Z.; Zhou, Y. Defense Strategy against Cyber Attacks on Substations Considering Attack Resource Uncertainty. *J. Mod. Power Syst. Clean Energy* **2025**, *13*, 1335–1346. [CrossRef]
2. Tanner, A.; Dancer, F.C.; Hall, J.; Parker, N.; Bishop, R.; McBride, T. The Need for Proactive Digital Forensics in Addressing Critical Infrastructure Cyber Attacks. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 976–982. [CrossRef]
3. Falowo, O.I.; Popoola, S.; Riep, J.; Adewopo, V.A.; Koch, J. Threat Actors' Tenacity to Disrupt: Examination of Major Cybersecurity Incidents. *IEEE Access* **2022**, *10*, 134038–134051. [CrossRef]
4. Kaloudi, N.; Li, J. The AI-Based Cyber Threat Landscape: A Survey. *ACM Comput. Surv.* **2020**, *53*, 20. [CrossRef]
5. Fieblinger, R.; Alam, M.T.; Rastogi, N. Actionable Cyber Threat Intelligence Using Knowledge Graphs and Large Language Models. In Proceedings of the 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Vienna, Austria, 8–12 July 2024; pp. 100–111. [CrossRef]
6. Sailio, M.; Latvala, O.-M.; Szanto, A. Cyber threat actors for the factory of the future. *Appl. Sci.* **2020**, *10*, 4334. [CrossRef]
7. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V. Advanced persistent threats (apt): Evolution, anatomy, attribution and countermeasures. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 9355–9381. [CrossRef]
8. Shahjee, D.; Ware, N. Integrated network and security operation center: A systematic analysis. *IEEE Access* **2022**, *10*, 27881–27898. [CrossRef]
9. Mavroeidis, V.; Hohimer, R.; Casey, T.; Jesang, A. Threat actor type inference and characterization within cyber threat intelligence. In Proceedings of the 2021 13th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 25–28 May 2021; pp. 327–352.
10. Radoglou-Grammatikis, P.; Kioseoglou, E.; Asimopoulos, D.; Siavvas, M.; Nanos, I.; Lagkas, T.; Argyriou, V.; Psannis, K.E.; Goudos, S.; Sarigiannidis, P. Surveying Cyber Threat Intelligence and Collaboration: A Concise Analysis of Current Landscape and Trends. In Proceedings of the 2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Naples, Italy, 4–6 December 2023; pp. 309–314. [CrossRef]
11. 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.
12. Al-Sada, B.; Sadighian, A.; Oligeri, G. MITRE ATT&CK: State of the Art and Way Forward. *ACM Comput. Surv.* **2024**, *57*, 12. [CrossRef]
13. Al-Sada, B.; Sadighian, A.; Oligeri, G. Analysis and characterization of cyber threats leveraging the MITRE ATT&CK database. *IEEE Access* **2023**, *12*, 1217–1234. [CrossRef]
14. Hooi, E.K.J.; Zainal, A.; Maarof, M.A.; Kassim, M.N. TAGraph: Knowledge Graph of Threat Actor. In Proceedings of the 2019 International Conference on Cybersecurity (ICoCSec), Porto, Portugal, 22–26 September 2019; pp. 76–80. [CrossRef]

15. Li, Y.; Liu, Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Rep.* **2021**, *7*, 8176–8186. [CrossRef]
16. Chng, S.; Lu, H.Y.; Kumar, A.; Yau, D. Hacker types, motivations and strategies: A comprehensive framework. *Comput. Hum. Behav. Rep.* **2022**, *5*, 100167. [CrossRef]
17. Alqudhaibi, A.; Albarrak, M.; Aloseel, A.; Jagtap, S.; Salonitis, K. Predicting cybersecurity threats in critical infrastructure for industry 4.0: A proactive approach based on attacker motivations. *Sensors* **2023**, *23*, 4539. [CrossRef] [PubMed]
18. Abou El Houda, Z. Cyber threat actors review: Examining the tactics and motivations of adversaries in the cyber landscape. In *Cyber Security for Next-Generation Computing Technologies*; CRC Press: Boca Raton, FL, USA, 2024; pp. 84–101.
19. Masmoudi, S. Unveiling the human factor in cybercrime and cybersecurity: Motivations, behaviors, vulnerabilities, mitigation strategies, and research methods. In *Cybercrime Unveiled: Technologies for Analysing Legal Complexity*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 41–91.
20. Bhole, M.; Sauter, T.; Kastner, W. Enhancing industrial cybersecurity: Insights from analyzing threat groups and strategies in operational technology environments. *IEEE Open J. Ind. Electron. Soc.* **2025**, *6*, 145–157. [CrossRef]
21. Thomas, M. Distinguishing cyberattacks by difficulty. *Int. J. Intell. CounterIntelligence* **2022**, *35*, 784–805. [CrossRef]
22. Singh, T. Case studies: State-Sponsored cyberattacks. In *Cybersecurity, Psychology and People Hacking*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 151–165.
23. Malik, V.; Khanna, A.; Sharma, N. Trends in ransomware attacks: Analysis and future predictions. *Int. J. Glob. Innov. Solut. (IJGIS)* **2024**. [CrossRef]
24. Folorunso, A.; Wada, I.; Samuel, B.; Mohammed, V. Security compliance and its implication for cybersecurity. *World J. Adv. Res. Rev.* **2024**, *24*, 2105–2121. [CrossRef]
25. Folorunso, A.; Mohammed, V.; Wada, I.; Samuel, B. The impact of ISO security standards on enhancing cybersecurity posture in organizations. *World J. Adv. Res. Rev.* **2024**, *24*, 2582–2595. [CrossRef]
26. Govea, J.; Gaibor-Naranjo, W.; Villegas-Ch, W. Transforming cybersecurity into critical energy infrastructure: A study on the effectiveness of artificial intelligence. *Systems* **2024**, *12*, 165. [CrossRef]
27. Snigdha, E.Z.; Jalil, M.S.; Dahwal, F.M.; Saeed, M.; Mehedy, M.T.J.; Hasan, S.K. Cybersecurity in Healthcare IT Systems: Business Risk Management and Data Privacy Strategies. *Am. J. Eng. Technol.* **2025**, *7*, 163–184. [CrossRef]
28. Hofbauer, J.; Mayer, K. Blue Team Fundamentals: Roles and Tools in a Security Operations Center. In Proceedings of the SECURWARE 2024: The Eighteenth International Conference on Emerging Security Information, Systems and Technologies, Woodbridge, VA, USA, 22–23 October 2024; pp. 176–184.
29. Zhang, J.; Bu, H.; Wen, H.; Liu, Y.; Fei, H.; Xi, R.; Li, L.; Yang, Y.; Zhu, H.; Meng, D. When llms meet cybersecurity: A systematic literature review. *Cybersecurity* **2025**, *8*, 55. [CrossRef]
30. Mahmoud, R.V.; Anagnostopoulos, M.; Pedersen, J.M. Detecting Cyber Attacks through Measurements: Learnings from a Cyber Range. *IEEE Instrum. Meas. Mag.* **2022**, *25*, 31–36. [CrossRef]
31. Mohammed, A. Transforming SOC Operations: Harnessing the Power of AI and ML for Enhanced Threat Detection. *Int. J. Res. Cult. Soc. Mon. Peer-Rev. Ref. Index.* **2024**, *8*. [CrossRef]
32. Baruwal Chhetri, M.; Tariq, S.; Singh, R.; Jalalvand, F.; Paris, C.; Nepal, S. Towards human-ai teaming to mitigate alert fatigue in security operations centres. *ACM Trans. Internet Technol.* **2024**, *24*, 12. [CrossRef]
33. Golushko, A.P.; Zhukov, V.G. Application of Advanced Persistent Threat Actors' Techniques aor Evaluating Defensive Countermeasures. In Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Russia, 27–30 January 2020; pp. 312–317. [CrossRef]
34. Yulianto, S.; Ngo, G.N.C. Automated Threat Hunting, Detection, and Threat Actor Profiling Using TIRA. In Proceedings of the 2024 International Conference on ICT for Smart Society (ICISS), Yogyakart, Indonesia, 4–5 September 2024; pp. 1–7. [CrossRef]
35. Grotto, A. Deconstructing Cyber Attribution: A Proposed Framework and Lexicon. *IEEE Secur. Priv.* **2019**, *18*, 12–20. [CrossRef]
36. Goel, S.; Nussbaum, B. Attribution Across Cyber Attack Types: Network Intrusions and Information Operations. *IEEE Open J. Commun. Soc.* **2021**, *2*, 1082–1093. [CrossRef]
37. Avellaneda, F.; Alikacem, E.H.; Jaafar, F. Using Attack Pattern for Cyber Attack Attribution. In Proceedings of the 2019 International Conference on Cybersecurity (ICoCSec), Negeri Sembilan, Malaysia, 25–26 September 2019; pp. 1–6. [CrossRef]
38. The MITRE Corporation. Techniques-Enterprise | MITRE ATT&CK®. Available online: https://attack.mitre.org/techniques/enterprise/ (accessed on 1 February 2026).
39. Althamir, M.A.; Boodai, J.Z.; Rahman, M.M.H. A Mini Literature Review on Challenges and Opportunity in Threat Intelligence. In Proceedings of the 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, 20–23 February 2023; pp. 558–563. [CrossRef]
40. Wu, Y.; Liu, Q.; Liao, X.; Ji, S.; Wang, P.; Wang, X.; Wu, C.; Li, Z. Price TAG: Towards Semi-Automatically Discovery Tactics, Techniques and Procedures OF E-Commerce Cyber Threat Intelligence. *IEEE Trans. Dependable Secur. Comput.* **2021**. *Early Access*. [CrossRef]

41. Agarwal, A.; Walia, H.; Gupta, H. Cyber Security Model for Threat Hunting. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021; pp. 1–8. [CrossRef]
42. Gopalakrishna, R.; Bhaskaran, R. Position Paper: The Extended Pyramid of Pain: The Attacker's Nightmare Edition. In *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*; IEEE: Piscataway, NJ, USA, 2025; pp. 662–670.
43. Awankar, P.; Kavathkar, D.; Ambekar, R.; Joy, A.; d Chandane, M. TTP-Based Dataset Generation and Classifier Model Training for Enhanced Cyber Threat Intelligence. *Procedia Comput. Sci.* **2026**, *283*, 5454–5463. [CrossRef]
44. Chen, Z.S.; Vaitheeshwari, R.; Wu, E.H.K.; Lin, Y.D.; Hwang, R.H.; Lin, P.C.; Lai, Y.C.; Ali, A. Clustering APT Groups Through Cyber Threat Intelligence by Weighted Similarity Measurement. *IEEE Access* **2024**, *12*, 141851–141865. [CrossRef]
45. Domschot, E.; Ramyaa, R.; Smith, M.R. Improving Automated Labeling for ATT&CK Tactics in Malware Threat Reports. *ACM Digit. Threat.* **2024**, *5*, 2. [CrossRef]
46. Permana, D.R.; Stiawan, D.; Rini, D.P.; Afifah, N.; Ningrum, S.K.; Budiarto, R. An Enhanced Method with Part of Speech Tagging and Named Entity Recognition Techniques Towards Advanced Persistent Threat in Cyber Threat Intelligence: Work in Progress. In Proceedings of the 2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 26–27 September 2024; pp. 493–498. [CrossRef]
47. Ramsdale, A.; Shiaeles, S.; Kolokotronis, N. A comparative analysis of cyber-threat intelligence sources, formats and languages. *Electronics* **2020**, *9*, 824. [CrossRef]
48. Huang, Y.T.; Vaitheeshwari, R.; Chen, M.C.; Lin, Y.D.; Hwang, R.H.; Lin, P.C.; Lai, Y.C.; Wu, E.H.K.; Chen, C.H.; Liao, Z.J.; et al. MITREtrieval: Retrieving MITRE Techniques From Unstructured Threat Reports by Fusion of Deep Learning and Ontology. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 4871–4887. [CrossRef]
49. Atlam, H.F. LLMs in Cyber Security: Bridging Practice and Education. *Big Data Cogn. Comput.* **2025**, *9*, 184. [CrossRef]
50. Saddi, V.R.; Gopal, S.K.; Mohammed, A.S.; Dhanasekaran, S.; Naruka, M.S. Examine the Role of Generative AI in Enhancing Threat Intelligence and Cyber Security Measures. In Proceedings of the 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 15–16 March 2024; pp. 537–542. [CrossRef]
51. Krašovec, A.; Steri, G.; Karopoulos, G.; Trapani, M. Large Language Models for Cyber Threat Intelligence: Extracting MITRE With LLMs. In Proceedings of the International Conference on Availability, Reliability and Security, Ghent, Belgium, 11–14 August 2025; pp. 80–89.
52. Gupta, M.; Akiri, C.; Aryal, K.; Parker, E.; Praharaj, L. From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access* **2023**, *11*, 80218–80245. [CrossRef]
53. Okey, O.D.; Udo, E.U.; Rosa, R.L.; Rodríguez, D.Z.; Kleinschmidt, J.H. Investigating ChatGPT and cybersecurity: A perspective on topic modeling and sentiment analysis. *Comput. Secur.* **2023**, *135*, 103476. [CrossRef]
54. Li, Y.; Shi, J.; Zhang, Z. An Approach for Rapid Source Code Development Based on ChatGPT and Prompt Engineering. *IEEE Access* **2024**, *12*, 53074–53087. [CrossRef]
55. Schmidt, D.C.; Spencer-Smith, J.; Fu, Q.; White, J. Towards a Catalog of Prompt Patterns to Enhance the Discipline of Prompt Engineering. *Ada Lett.* **2024**, *43*, 43–51. [CrossRef]
56. Tripaldelli, A.; Pozek, G.; Butka, B. Leveraging Prompt Engineering on ChatGPT for Enhanced Learning in CEC330: Digital System Design in Aerospace. In Proceedings of the 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 8–11 May 2024; pp. 1–9. [CrossRef]
57. Xiang, X.; Liu, H.; Zeng, L.; Zhang, H.; Gu, Z. IPAttributor: Cyber attacker attribution with threat intelligence-enriched intrusion data. *Mathematics* **2024**, *12*, 1364. [CrossRef]
58. Noor, U.; Anwar, Z.; Rashid, Z. An Association Rule Mining-Based Framework for Profiling Regularities in Tactics Techniques and Procedures of Cyber Threat Actors. In Proceedings of the 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), Shah Alam, Malaysia, 11–12 July 2018; pp. 1–6. [CrossRef]
59. Desa, J.M.; Juremi, J.; Jenalis, M.H. Understanding Cyber Threat Actors: A Review on Classification, Motivations, and Behavioral Profiling. In *International Conference on Data Engineering and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 23–30.
60. Abo-Alian, A.; Youssef, M.; Badr, N.L. A data-driven approach to prioritize MITRE ATT&CK techniques for active directory adversary emulation. *Sci. Rep.* **2025**, *15*, 27776. [CrossRef] [PubMed]
61. Duan, J.; Luo, Y.; Zhang, Z.; Peng, J. A heterogeneous graph-based approach for cyber threat attribution using threat intelligence. In Proceedings of the ACM International Conference on Machine Learning and Computing, Shenzhen, China, 2–5 February 2024; pp. 87–93. [CrossRef]
62. Kida, M.; Olukoya, O. Nation-State Threat Actor Attribution Using Fuzzy Hashing. *IEEE Access* **2023**, *11*, 1148–1165. [CrossRef]
63. Xiao, N.; Lang, B.; Wang, T.; Chen, Y. APT-MMF: An advanced persistent threat actor attribution method based on multimodal and multilevel feature fusion. *Comput. Secur.* **2024**, *144*, 103960. [CrossRef]

64. Irshad, E.; Basit Siddiqui, A. Cyber threat attribution using unstructured reports in cyber threat intelligence. *Egypt. Inform. J.* **2023**, *24*, 43–59. [CrossRef]
65. Irshad, E.; Siddiqui, A.B. Context-aware cyber-threat attribution based on hybrid features. *ICT Express* **2024**, *10*, 553–569. [CrossRef]
66. Saha, A.; Blasco, J.; Cavallaro, L.; Lindorfer, M. Adapt it! automating apt campaign and group attribution by leveraging and linking heterogeneous files. In Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses, Padua, Italy, 30 September–2 October 2024; pp. 114–129.
67. Saha, A.; Mattei, J.; Blasco, J.; Cavallaro, L.; Votipka, D.; Lindorfer, M. Expert Insights into Advanced Persistent Threats: Analysis, Attribution, and Challenges. In Proceedings of the 34th USENIX Security Symposium (USENIX Sec), Seattle, WA, USA, 13–15 August 2025.
68. Hou, C.; Xie, W.; Li, F.; Meng, X.; Shi, Y.; Liu, Y.; Zhao, Z. Data Pattern Matching Method Based on BERT and Attention Mechanism. In Proceedings of the 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, 9–10 August 2024; pp. 1–8. [CrossRef]
69. Duan, L.; Wen, M.; Xiong, Y. MLDSJ: A multi-level feature joint attribution method for APT group based on threat intelligence. *EURASIP J. Inf. Secur.* **2025**, *20*. [CrossRef]
70. Ge, W.; Wang, J.; Cui, Z.; Fang, Z.; Zhan, W. ThreatMAMBA: Achieving High-Robustness Cyber Threat Attribution During the Evolution of Attacks. *IEEE Trans. Inf. Forensics Secur.* **2026**, *21*, 4386–4401. [CrossRef]
71. Böge, E.; Ertan, M.B.; Alptekin, H.; Çetin, O. Unveiling Cyber Threat Actors: A Hybrid Deep Learning Approach for Behavior-Based Attribution. *ACM Digit. Threat.* **2025**, *6*, 2. [CrossRef]
72. Basnet, A.S.; Ghanem, M.C.; Dunsin, D.; Kheddar, H.; Sowinski-Mydlarz, W. Advanced persistent threats (apt) attribution using deep reinforcement learning. *Digit. Threat. Res. Pract.* **2025**, *6*, 14. [CrossRef]
73. Beltrán, P.; Gil Pérez, M.; Nespoli, P. Antifragile Cyber Deception: Profiling Threat Actors Through Adaptive Intelligence and Semantic Inference. Available online: https://ssrn.com/abstract=5598038 (accessed on 1 June 2026).
74. Zhang, Y.; Yang, P.; Jiang, Z.; Ma, C.; Cui, M.; You, Y. APTChaser: Cyber Threat Attribution via Attack Technique Modeling. In *International Conference on Digital Forensics and Cyber Crime*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 168–185.
75. N, S.; Puzis, R.; Angappan, K. Deep Learning for Threat Actor Attribution from Threat Reports. In Proceedings of the 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), Nagoya, Japan, 26–29 June 2020; pp. 1–6. [CrossRef]
76. Rani, N.; Saha, B.; Shukla, S.K. A comprehensive survey of automated Advanced Persistent Threat attribution: Taxonomy, methods, challenges and open research problems. *J. Inf. Secur. Appl.* **2025**, *92*, 104076. [CrossRef]
77. Geetha, R.; Thilagam, T. A Review on the Effectiveness of Machine Learning and Deep Learning Algorithms for Cyber Security: R. Geetha, T. Thilagam. *Arch. Comput. Methods Eng.* **2021**, *28*, 2861–2879.
78. Ozkan-Okay, M.; Akin, E.; Aslan, Ö.; Kosunalp, S.; Iliev, T.; Stoyanov, I.; Beloev, I. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. *IEEE Access* **2024**, *12*, 12229–12256. [CrossRef]
79. Shwartz-Ziv, R.; Armon, A. Tabular data: Deep learning is not all you need. *Inf. Fusion* **2022**, *81*, 84–90. [CrossRef]
80. Mohale, V.Z.; Obagbuwa, I.C. A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity. *Front. Artif. Intell.* **2025**, *8*, 1526221. [CrossRef] [PubMed]
81. Zettl-Schabath, K.; Bund, J.; Müller, M.; Borrett, C.; Hemmelskamp, J.; Alibegovic, A.; Bajra, E.; Jazxhi, A.; Kellenter, E.; Sachs, A.; et al. Global Dataset of Cyber Incidents (Version 1.3.2). Available online: https://zenodo.org/records/14965395 (accessed on 14 March 2025).
82. European Repository of Cyber Incidents. About Us-EuRepoC: European Repository of Cyber Incidents. Available online: https://eurepoc.eu/about-us/ (accessed on 1 March 2025).
83. Vićić, J.; Baram, G.; Gartzke, E. What lurks beneath the tip of the iceberg? Exploring the "missingness" problem in cyber events data. *Int. Interact.* **2026**, *52*, 185–212. [CrossRef]
84. Baram, G. Re-ordering accountability: The significance of joint public attribution in a fragmented cyberspace. *Contemp. Secur. Policy* **2026**. [CrossRef]
85. Aybar, P.F. Gueretational choices in cyberspace: Quantitative insights into targeting and attack behavior among cyber offenders. *J. Crim. Justice* **2026**, *103*, 102621. [CrossRef]
86. Syed, A.; Nour, B.; Pourzandi, M.; Assi, C.; Debbabi, M. Comprehensive Advanced Persistent Threats Dataset. *IEEE Netw. Lett.* **2025**, *7*, 150–154. [CrossRef]
87. Xu, H.; Wang, S.; Li, N.; Wang, K.; Zhao, Y.; Chen, K.; Yu, T.; Liu, Y.; Wang, H. Large language models for cyber security: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.* **2024**. [CrossRef]
88. Sewunetie, W.T.; Kovács, L. Exploring Sentence Parsing: OpenAI API-Based and Hybrid Parser-Based Approaches. *IEEE Access* **2024**, *12*, 38801–38815. [CrossRef]

89. Kulkarni, N.D.; Tupsakhare, P. Strategies for avoiding GPT hallucinations. *Int. Res. J. Mod. Eng. Technol. Sci.* **2024**, *6*, 60718. [CrossRef]
90. Pagani, P. Russian APT28 Hacker Accused of the NATO Think Tank Hack in Germany. Available online: https://securityaffairs.com/132452/hacking/apt28-hacked-nato-think-tank.html (accessed on 7 February 2026).
91. Church, K. Emerging trends: When can users trust GPT, and when should they intervene? *Nat. Lang. Eng.* **2024**, *30*, 417–427. [CrossRef]
92. Dahouda, M.K.; Joe, I. A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access* **2021**, *9*, 114381–114391. [CrossRef]
93. Alketbi, K.S.; Mehmood, A. A comprehensive survey of explainable artificial intelligence techniques for malicious insider threat detection. *IEEE Access* **2025**, *13*, 121772–121798. [CrossRef]
94. Qian, C. Web Malicious Request Identification Based on k-Nearest Neighbor Algorithm. In Proceedings of the 2024 7th International Conference on Computer Information Science and Application Technology (CISAT), Hangzhou, China, 12–14 July 2024; pp. 131–135. [CrossRef]
95. Dolai, S.; Kumari, A.; Agarwal, M. SANVector: SBERT-APTNet Vector Framework for Cyber Threat Attack Attribution Using Diversified CTI Logs. In Proceedings of the International Conference on Information Systems Security, Indore, India, 16–20 December 2025; pp. 136–148.
96. JetBrains. PyCharm: The Only Python IDE You Need. Available online: https://www.jetbrains.com/pycharm/ (accessed on 10 May 2026).
97. Rahman, A.; Hassan, I.; Ahad, M.A.R. Nurse Care Activity Recognition: A Cost-Sensitive Ensemble Approach to Handle Imbalanced Class Problem in the Wild. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Virtual, 21–26 September 2021; pp. 440–445. [CrossRef]
98. Huseynli, A. Cyber Threat Actor Attribution: A Systematic Review and Evolutionary Perspective. *Int. J. Inf. Secur. Sci.* **2025**, *15*, 29–43. [CrossRef]
99. Tufail, S.; Riggs, H.; Tariq, M.; Sarwat, A.I. Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. *Electronics* **2023**, *12*, 1789. [CrossRef]
100. Cunningham, P.; Delany, S.J. K-nearest neighbour classifiers-a tutorial. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 128. [CrossRef]
101. Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *J. Big Data* **2024**, *11*, 113. [CrossRef]
102. Cervantes, J.; Garcia Lamont, F.; Rodriguez Mazahua, L.; Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **2020**, *408*, 189–215. [CrossRef]
103. Zhou, Y.; Liu, P.; Qiu, X. KNN-contrastive learning for out-of-domain intent classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 5129–5141.
104. Del Moral, P.; Nowaczyk, S.; Pashami, S. Why is multiclass classification hard? *IEEE Access* **2022**, *10*, 80448–80462. [CrossRef]
105. Zidan, K.; Alam, A.; Allison, J.; Al-Sherbaz, A. Assessing the challenges faced by Security Operations Centres (SOC). In Proceedings of the Future of Information and Communication Conference, Berlin, Germany, 4–5 April 2024; pp. 256–271.
106. Yadav, S. Social automation and APT attributions in national cybersecurity. *J. Cyber Secur. Technol.* **2024**, *8*, 286–311. [CrossRef]
107. Egloff, F.J.; Cavelty, M.D. Attribution and Knowledge Creation Assemblages in Cybersecurity Politics. *J. Cybersecur.* **2021**, *7*. [CrossRef]
108. Ainslie, S.; Thompson, D.; Maynard, S.; Ahmad, A. Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Comput. Secur.* **2023**, *132*, 103352. [CrossRef]