applied sciences

MDPI

Article

CAVeCTIR: Matching Cyber Threat Intelligence Reports on Connected and Autonomous Vehicles Using Machine Learning †

George E. Raptis [1,*], Christina Katsini [1], Christos Alexakos [1,*], Athanasios Kalogeras [1] and Dimitrios Serpanos [2,3]

[1] Industrial Systems Institute (ISI), Athena Research & Innovation Center (ATHENA), 26504 Patras, Greece
[2] Electrical & Computer Engineering, University of Patras, 26504 Patras, Greece
[3] Computer Technology Institute and Press "Diophantus", University Campus of Patras, 26504 Patras, Greece
* Correspondence: graptis@isi.gr (G.E.R.); alexakos@isi.gr (C.A.)
† This paper is an extended version of our paper published in Proceedings of the 2021 IEEE International Conference on Cyber Security and Esilience (CSR), Virtual, 26–28 July 2021.

Abstract: Connected and automated vehicles (CAVs) are getting a lot of attention these days as their technology becomes more mature and they benefit from the Internet-of-Vehicles (IoV) ecosystem. CAVs attract malicious activities that jeopardize security and safety dimensions. The cybersecurity systems of CAVs detect such activities, collect and analyze related information during and after the activity, and use cyber threat intelligence (CTI) to organize this information. Considering that CTI collected from various malicious activities may share common characteristics, it is critical to provide the cybersecurity stakeholders with quick and automatic ways of analysis and interrelation. This aims to help them perform more accurate and effective forensic investigations. To this end, we present CAVeCTIR, a novel approach that finds similarities between CTI reports that describe malicious activities detected on CAVs. CAVeCTIR uses advanced machine learning techniques and provides a quick, automated, and effective solution for clustering similar malicious activities. We applied CAVeCTIR in a series of experiments investigating almost 3000 malicious activities in simulation, real-world, and hybrid CAV environments, covering seven critical cyber-attack scenarios. The results showed that the DBSCAN algorithm identified seven no-overlapping core clusters characterized by high density. The results indicated that cybersecurity stakeholders could take advantage of CAVeCTIR by adopting the same or similar methods to analyze newly detected malicious activity, speed up the attack attribution process, and perform a more accurate forensics investigation.

Keywords: connected and autonomous vehicles; internet of vehicles; cyber threat intelligence reports; cybersecurity; machine learning; cluster analysis; malicious incidents and attacks; security response; threat profiling and information sharing; digital forensics

Citation: Raptis, G.E.; Katsini, C.; Alexakos, C.; Kalogeras, A.; Serpanos, D. CAVeCTIR: Matching Cyber Threat Intelligence Reports on Connected and Autonomous Vehicles Using Machine Learning. *Appl. Sci.* **2022**, *12*, 11631. https://doi.org/10.3390/app122211631

Academic Editor: Jose Machado

Received: 13 October 2022
Accepted: 10 November 2022
Published: 16 November 2022

1. Introduction

Recent and rapid technological advancements have resulted in building smart and complex environments (e.g., smart cities). Among others, smart mobility (i.e., intelligent transport and mobility networks) is an important part of such environments. Smart mobility refers to many modes of transport, with connected and autonomous vehicles (CAVs) being a rising and demanding domain. CAVs aims to assist driving activity by combining various technologies (e.g., advanced sensors, onboard computing, remote processing, telecommunication, and positioning systems) through automated procedures. To accomplish their goal, CAVs collect and analyze data from their surrounding context continuously and navigate within the smart environment with little or no human intervention. CAVs operate on Internet-of-Vehicles (IoV) networks, which are dynamic and multi-diverse ecosystems that include various communication types. Examples of such communication types are Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Everything (V2X).

IoV is a challenging domain with complex and diverse features and demands, such as dynamic topological structures, huge network scalability, non-uniform distribution of network components, complex granularities, and mobility limitations [1]. In this context, CAVs are an IoV dimension that attracts diverse types of malicious activities (e.g., cyber-attacks). We identify three main types of such activities: (i) malicious activities on autonomous control systems (e.g., in-vehicle network attacks), (ii) malicious activities on autonomous driving system components (e.g., sensor attacks), and (iii) malicious activities on V2X communications (e.g., attacks on ad-hoc vehicle networks) [2]. Their successful confrontation is critical, considering that vulnerabilities in CAVs can put human life at risk.

Researchers have proposed several defense techniques to identify malicious activities [3] and apply response and recovery strategies [4]. During the application of such techniques, information (e.g., characteristics of the detected malicious activity, threat actor profiling, and existing techniques that confront cyber-attacks) is collected and analyzed. Such information is critical for the successful mitigation of potentially harmful activities and incidents in CAVs. The structure of such information should follow standard procedures to be beneficial for security and public safety within a smart environment (e.g., IoV ecosystems). Toward this direction, the cybersecurity community has proposed the use of Cyber Threat Intelligence (CTI) reports. In recent years, CTI has attracted growing attention and investment from cybersecurity communities (both research and industry), considering that cybersecurity stakeholders use CTI to build accessible knowledge repositories, which contain rich information about identified malicious activities, detected cyber-attacks, threat profiling, analysis methods, adversary tactics and techniques, forensics investigation, etc. Cybersecurity stakeholders (e.g., cybersecurity experts) use such knowledge to speed up the analysis process of the detected malicious activities and eventually attribute them successfully, aiming to mitigate their direct impact and prevent similar malicious activities in the future.

Focusing on cyber-attacks, recent research works take advantage of artificial intelligence (e.g., machine learning processes) to detect high-level indicators of compromise (IOCs), which are presented in CTI reports, to enable attack attribution [5]. Furthermore, researchers have proposed intelligent systems that model CTI and detect threat types (e.g., anomaly types) on heterogeneous information networks [6]. In an attempt to follow and broaden these strategies, cybersecurity stakeholders could use CTI-based solutions to minimize false alerts, fatigue, and misdetections. Therefore, cybersecurity stakeholders could enrich the sharing of IOCs with machine learning models to achieve collaborative threat detection [7]. Such research works use varying CTI platforms and repositories to store information related to threats, IOCs, and machine learning models, such as MISP, GOSINT, YETI, and CIF [8].

Even though there is much information, it is scattered throughout several sources, such as publicly accessible knowledge bases (e.g., MITRE ATT&CK) and private CTI repositories within an organization. Cybersecurity stakeholders, with an emphasis on cybersecurity experts, should be able to rapidly identify common patterns and characteristics in CTI reports (e.g., common adversary tactics, similar analysis and mitigation techniques), given that the detected malicious activities (e.g., cyber-attacks) may be connected to one another. Therefore, there is a need to assist cybersecurity stakeholders in this direction, as this would have a direct impact on the forensic investigation process (e.g., speed up the attack attribution process and perform a more accurate post-incident investigation). Motivated by that need, we can extend existing cybersecurity frameworks for IoV, such as nIoVe [9], by reinforcing them with tools that employ artificial intelligence techniques to address such challenges. In particular, we can adopt machine-learning approaches (e.g., clustering techniques) to build tools that identify similarities and differences between the attributes of the stored CTI reports and the characteristics of a new malicious activity. Cybersecurity experts could use such tools to swiftly correlate a malicious activity with previously reported ones, access existing knowledge about it, and apply comparable approaches to analyze its impact. This would result in a more accurate and quicker attack attribution and

post-incident forensics investigation. Therefore, the research question that this work aims to answer is:

> *"Can we expand existing IoV cybersecurity frameworks and provide the cybersecurity stakeholders (with an emphasis on cybersecurity experts) with an approach implementing an automatic process that analyzes and correlates CTI reports with newly detected malicious activities (e.g., cyber-attacks) on CAVs in an IoV environment?"*

To answer our research question, we developed CAVeCTIR, which expands the nIoVe framework [9]. CAVeCTIR uses advanced machine learning techniques and provides a quick, automated, and effective solution for clustering similar malicious activities. We performed an evaluation study applying CAVeCTIR in various CAV environments (simulation, real-world, and hybrid) and investigated its performance for almost 3000 malicious activities of diverse types (GNSS signal spoofing, GNSS signals jamming, CAN DoS attack, alteration of CAN messages, alteration of camera stream, false sensor reading, and malware/ransomware attack). The results indicated that the DBSCAN algorithm worked well and identified non-overlapping core clusters of high density. Therefore, cybersecurity stakeholders could take advantage of CAVeCTIR by adopting the same or similar methods to analyze newly detected malicious activity, speed up the attack attribution process, and perform a more accurate forensics investigation.

The remainder of the paper is organized as follows: we discuss related works, we present the nIoVe framework, which is a multi-layered interoperable cybersecurity solution for IoV [9], we present our approach for matching CTI reports using machine learning techniques, we present the setup and the results of an evaluation study with multiple attack scenarios, we interpret and discuss the findings of our study, we present the future directions of this work, and we conclude the paper.

2. Related Work

In this section, we discuss related works in the field focusing on the importance of CTI in cybersecurity frameworks, methods for extracting and sharing CTI information, and the analysis methods of CTI for IoV systems.

2.1. Importance of CTI

CTI provides meaningful information about cybersecurity threats; through the analysis of such information, we build valuable knowledge about the motives, capabilities, resources, and opportunities of cyberspace adversaries. Such knowledge is important for organizations that focus on providing cybersecurity solutions and individuals that serve as network architects, cybersecurity operators, forensic specialists, threat incident responders, policy-makers, etc. Several challenges (e.g., attack vector reconnaissance, attack indicator reconnaissance) and opportunities (e.g., application of artificial intelligence techniques to perceive, reason, learn, and act against advanced cyber-attacks) can be identified in the CTI [10].

CTI is a critical component of the cybersecurity domain because it provides evidence-based knowledge about existing and potential cyber threats. It also improves security operations efficiency and effectiveness in terms of detective and preventive capabilities. To represent the knowledge gained through CTI, several taxonomies, sharing standards, and ontologies have been proposed (e.g., [11–13]), which typically address the following aspects: motivations, goals, strategies, tactics, techniques, and procedures (TTPs), tools, indicators of compromise (IOCs), atomic indicators, targets, and courses of action. Multiple attacks and actors are supported through such CTI models.

Following the construction of the CTI information, its quality assessment is also an important step. Several levels of subjective and objective assessment can be identified, including attribute (e.g., concise representation, relevancy, schema completeness, syntactic accuracy, timeliness), object (e.g., representational consistency, reputation), and report (e.g., the appropriate amount of data) levels. In their recent work, Schlette et al. [14] proposed quality assessment and measurement methods, also discussing visualization

techniques. Besides the steps of CTI reconstruction and quality assessment, it is also important for cybersecurity organizations and individuals to follow novel and consistent ways of extracting and sharing CTI information. We discuss them next.

2.2. Extracting and Sharing CTI Information

Recent research works make extensive use of machine-learning approaches to extract CTI information, which has various advantages, such as the identification of indicators of compromise and automatic categorization of CTI reports with domain-specific tags (e.g., finance, government). Various methods have been used, such as analysis of raw log data [15], analysis of social media data based on convolutional neural networks [16], analysis of conversations (e.g., forum data) based on support vector machines [17,18], use of natural language processing models [19], analysis of semantic and contextual information collected from varying sources [20], mining of CTI reports to automatically learn the semantics of malicious campaigns [21], and leveraging entropy and mutual information [22], from structured or unstructured sources [23,24]. A common characteristic of these works is that they rely on artificial intelligence and that the analysis of the CTI information is typically organized (e.g., in CTI reports), which is then shared among cybersecurity stakeholders.

Regarding the sharing of CTI information, Wagner et al. [25], in a recent review, showed that various sharing models are adopted, covering peer-to-peer, peer-to-repository, and hybrid aspects. Considering the importance of actionable CTI and the value of various dimensions (e.g., timeliness of sharing CTI, trust establishment of CTI sharing, stakeholder reputation, data interoperability, privacy, and anonymity), the use of automated tools (e.g., through established CTI repositories) plays a major role. One of the most known and widely used tools is the MISP Threat Sharing platform [26], which is an open-source solution that supports utilities and documentation for effective CTI by sharing indicators of compromise. Other tools, such as SecurityKG [27], are also used for automated CTI gathering and management, supporting higher-level concepts (e.g., adversary tactics, techniques, and procedures). Moreover, organizations often use technologies such as STIX [12] and TAXII [28] to describe and share CTI information in a secure and automated manner.

2.3. Use and Analysis of CTI in IoV Systems

A small body of research focuses on the use and analysis of CTI on IoV, which might result from the fact that IoV technologies are new, are not yet mature, and have not been in wide practical use until now. Kukkala et al. [29] underpin the importance of using threat intelligence for enhanced cybersecurity testing and sharing this information between different organizations to effectively tackle cyber-attacks in CAVs and IoV systems. Along the same line, He et al. [30] highlight the benefits of adopting artificial intelligence techniques (e.g., machine learning) to analyze CTI in IoV systems, such as dealing with a huge amount of data and performing automated processes. Panda et al. [31] proposed that honeypots could be adjusted to deceive attackers and thus collect and analyze CTI in IoV contexts, configuring the IoV vulnerabilities. Basnet et al. [32] present deep-learning techniques that could leverage CTI for cyber-attack detection in CAVs. Ali et al. [33] proposed the adoption of machine-learning approaches for secure vehicular communication. Other important dimensions, apart from security, are preservation and safety. Regarding preservation, Liu et al. [34] proposed an IoV service deployment and execution with privacy preservation in cloud-edge computing. Regarding safety, Mohseni et al. [35] reviewed practical machine-learning safety techniques that could complement engineering safety for machine-learning-based software in autonomous vehicles. In a related domain (maritime transportation systems), Kumar et al. [36] presented a framework that models CTI and identifies domain-specific threat types.

2.4. Synthesis

The discussion of the related works in the field (Table 1) highlighted the importance of CTI in cybersecurity frameworks, presented methods for extracting [15–17,19–24] and sharing [12,26–28] CTI information, and stressed that only a limited body of research exists in the intersection of CTI and IoV [29–32]. To this end, we should stress that while the analysis (e.g., discovery of similarities and differences, comparisons) and correlation (e.g., matching) of CTI reports is an important task; there is no research—to the authors' knowledge—that elaborates on it. Hence, aiming to shed light on this dimension and answer our research question, in the next sections, we discuss the nIoVe framework that provides a cybersecurity framework focusing on IoV; we propose a novel approach based on machine learning that supports the automatic analysis and correlation of CTI reports, and present the results of an evaluation study.

Table 1. Related works.

Work	Reason(s) for Adopting Machine Learning Techniques	Application Domain
Landauer et al. [15]	extract CTI from raw log data	generic
Zhao et al. [16]	extract CTI from social data	domain-specific
Deliu et al. [17]	classify CTI content (forum posts)	social media
Zhang et al. [19]	extract CTI actions through semantic analysis	generic, text mining
Li et al. [20]	collect and analyze CTI events (semantic analysis of articles)	generic, text mining
Zhu et al. [21]	collect and analyze CTI malicious campaigns (semantic analysis of articles)	generic, text mining
Husari et al. [22]	extract CTI actions through semantic analysis	generic, text mining
Ghazi et al. [23]	extract high-level CTI IOCs	generic, text mining
Wang et al. [24]	extract high-level CTI IOCs	generic, text mining
Gao et al. [27]	search, collect, and manage CTI	generic
He et al. [30]	analyze huge CTI volume	IoV
Panda et al. [31]	collect and analyze CTI for honeypot configuration	IoV
Basnet et al. [32]	detect cyber-attacks in CAVs	IoV
Ali et al. [33]	ensure secure vehicular communication	IoV
Liu et al. [34]	deploy and execute privacy preservation services in cloud-edge computing	IoV
Mohseni et al. [35]	complement engineering safety for IoV software based on machine learning	IoV

3. nIoVe Framework

Our work is based on the nIoVe framework [9], which provides a generalized cybersecurity solution for IoV networks, focusing on CAVs. A critical feature of any integrated cyber-defense strategy, such as the ones described in nIoVe, is the real-time and post-incident investigation of the incoming attacks. This two-phase investigation approach aims to identify characteristics, motives, targeted assets, and the impact of the identified attacks. Security stakeholders, such as security experts, could take advantage of the investigation outcome to have a better understanding of the underlying causes of the incoming attacks and identify common patterns (e.g., attacker profiling, exploitation of IoV vulnerabilities). Toward this end, nIoVe implements an attack attribution and digital forensic readiness tool (AAFRT) to allow security experts to perform post-incident analysis of the detected cyber-attacks [37]. AAFRT supports two major functionalities:

- *Attack attribution*, which identifies low-level and high-level IOCs and aims to assist security experts in attributing the detected attack to known threat actors by attempting to identify common tactics, techniques, and procedures;
- *Digital forensic readiness*, which enables the nIoVe framework to automatically collect forensic data from various sources and to create a CTI report containing all the necessary information about the attack. This step is vital for further digital investigation, preservation of the integrity of the collected data, and maintenance of a valid chain of custody.

In the IoV environments, there is a large degree of heterogeneity for the connected devices that handle diverse information types and communicate through different protocols.

That is a major challenge, and security experts address it by building customized forensics plans. Such plans allow the automatic collection and analysis of data related to the detected cyber-attack, aiming at mitigating the system vulnerability. Moreover, the nIoVe framework communicates the identified vulnerabilities both to the administration/monitoring team and to the manufacturers of the IoV components (e.g., manufacturers of CAVs).

To this end, AAFRT provides the FoRePlan tool [38] that allows security experts to create new and update old forensics plans tailored to the type of supported attacks (e.g., denial-of-service, GNSS spoofing attacks). After the detection of the attack, the plan that matches the features of the attack is selected, scheduled, and executed. A multi-threading process initiates the collection of related data from various IoV sources. Next, standardized methods and procedures are followed to preserve the collected data. A preliminary analysis (e.g., timeline analysis, session analysis, log analysis) of the data follows to speed up the forensics analysis process. After collecting all the available information, AAFRT provides the security experts with a CTI report with several characteristics related to the detected attack, including the assigned attack profile, the executed plan, the collected data, and the outcome of the forensics analysis. Moreover, nIoVe provides security experts with a secure virtual environment equipped with sophisticated forensics analysis tools to help them with the analysis process.

CAVeCTIR Method

As discussed, the attack attribution process and the appropriate assignment of an attack profile are crucial steps in post-incident forensic analysis. Some characteristics and features of the detected attack can be directly extracted from the analysis of the acquired data. Others require more intense and complex procedures of processing and knowledge extraction, which are based on interrelating different CTI reports. These CTI reports are recorded in a CTI repository based on the MISP sharing platform [26]. The CTI reports might share common characteristics (e.g., attack techniques, attack models, analysis methods), and thus, they might be potentially correlated with each other. However, this information is not visible to the cybersecurity stakeholders, and thus, we propose the CAVeCTIR method, which supports the automatic interrelation and matching of a detected malicious activity with a collection of stored CTI reports. CAVeCTIR extends the CTIMatcher tool presented in our previous work [39], as it provides a more holistic solution that supports more and diverse types of malicious activities found in IoV. The CAVeCTIR aims to assist cybersecurity stakeholders (e.g., cybersecurity experts and analysts) in identifying CTI reports that describe similar malicious activities and thus boost post-incident forensic analysis. CAVeCTIR supports the following processes (Figures 1 and 2), which are technically implemented through micro-services in the IoV ecosystem.

- *Initialization of attack profile*: CAVeCTIR extracts characteristics of the detected attack directly and initializes the attack profile. These characteristics include the IoV components (e.g., sensors) that have been attacked, propagation trends, risk assessment regarding the overall IoV system, and the level of confidence regarding the impact of the attack. Based on these, an initial attack profile is generated and assigned to a new forensic case.
- *Acquisition of CTI reports*: In parallel with the previous step, CAVeCTIR seeks, processes, and acquires CTI reports from various platforms and repositories, including both the internal MISP-based CTI repository and other external repositories to which the organization has access (e.g., shared threat intelligence repository). The output of this step is a collection of the acquired CTI reports.
- *Features selection/extraction*: Based on the collected information related to the detected attack and the CTI reports, CAVeCTIR aims to reduce the number of features and rank the importance of these features through feature selection and extraction processes. It generates a features model fed to the matching analysis process, aiming to provide an efficient and quick machine-learning clustering approach.

- *Algorithm selection*: Several algorithms are supported for both supervised and unsupervised machine learning. Focusing on the unsupervised techniques, the supported algorithms include k-means clustering, mean-shift clustering, density-based spatial clustering of applications with noise (DBSCAN), affinity propagation, hierarchical clustering, balanced iterative reducing and clustering using hierarchies (BIRCH), and ordering points to identify the clustering structure (OPTICS). The selection of the most effective algorithm depends on various factors, including the selected features, the knowledge of the number of clustering, and the sample size.
- *Matching analysis*: Having as input the profile of the detected attack, the acquired CTI reports, the features model, and the appropriate machine-learning algorithm, CAVeCTIR proceeds with the matching analysis, which aims to cluster the CTI reports dynamically and in real-time. Similarities and differences between the CTI reports are recorded. In the end, the detected attack is matched with one or more CTI reports, which are characterized as interrelated, meaning that they describe malicious activities, tools, techniques, and processes that share common patterns with the newly detected attack.
- *Recommendation of matched CTI reports*: The outcome of the previous step is a series of clusters containing matched recorded CTI reports. Based on the detected attack, CAVeCTIR recommends the corresponding cluster to the cybersecurity stakeholders (e.g., cybersecurity experts and analysts). This cluster contains CTI reports, from both internal and external sources, that match the profile of the detected attack, according to the selected machine-learning approach. Therefore, the cybersecurity stakeholders have full access to the matched CTI reports, assess their relevance to the detected attack, estimate the impact on the IoV environment, and can adopt the same or similar post-incident forensic analysis processes (including tools, methods, and techniques) to identity and eventually attribute the detected IoV attack quickly and accurately.

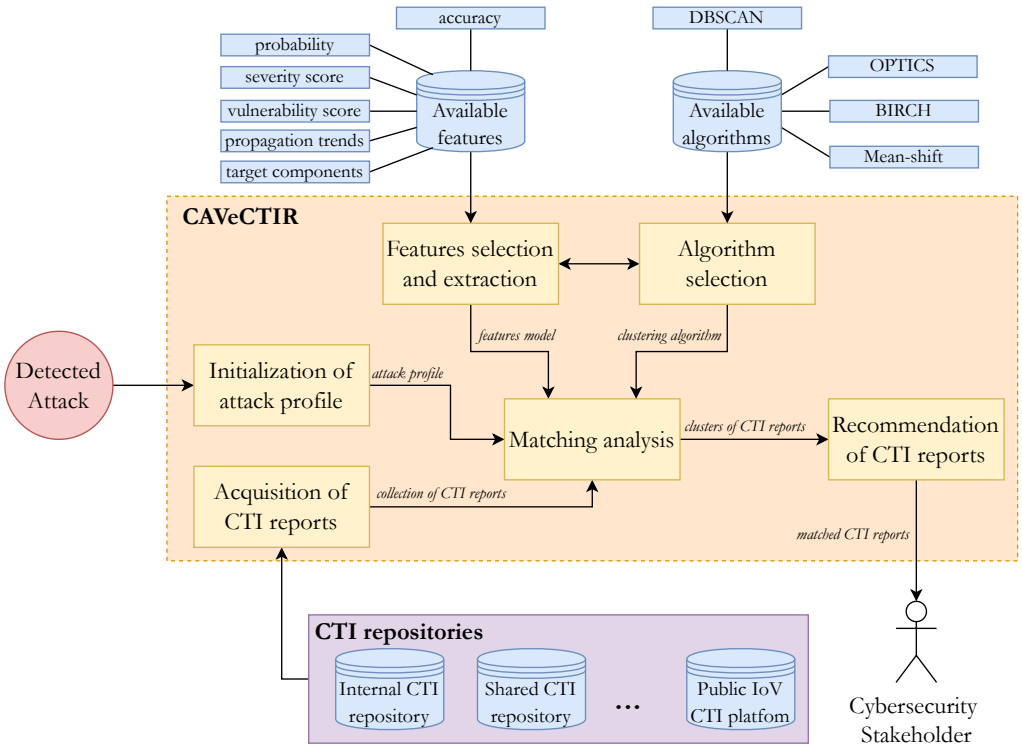


Figure 1. The conceptual structural diagram of CAVeCTIR.

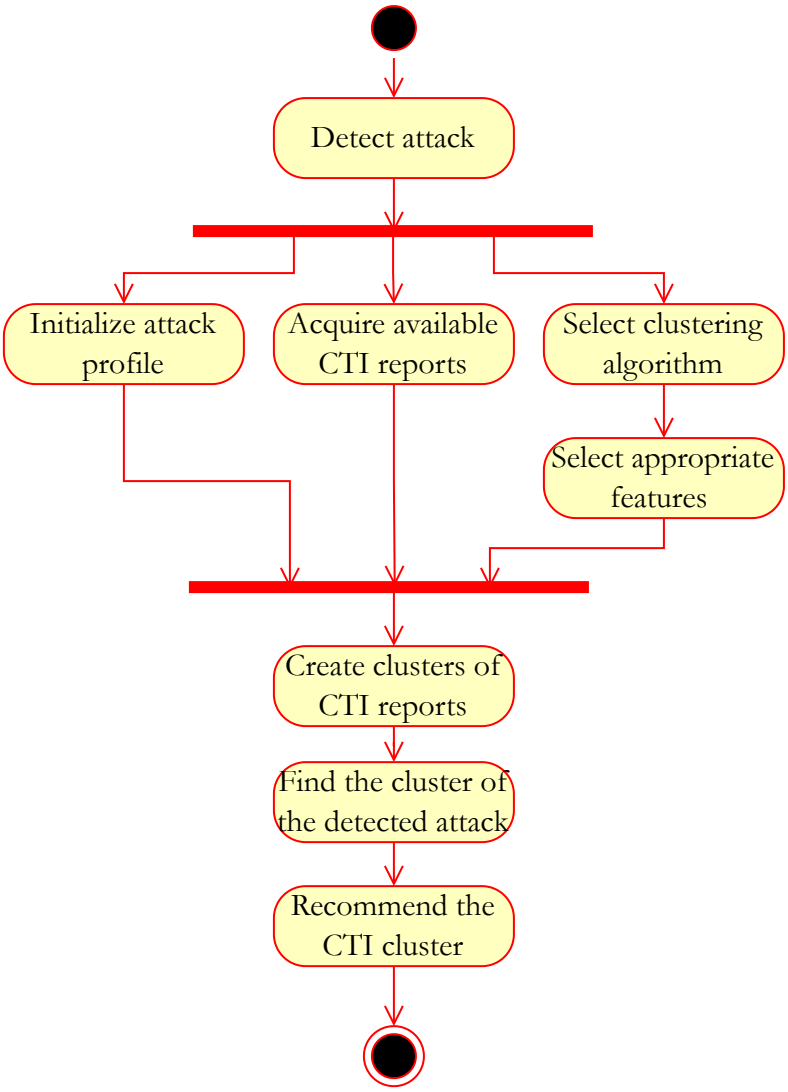


Figure 2. The activity diagram of the primary functionality (behavior) of CAVeCTIR.

4. Evaluation Study

After proposing an automatic process for analyzing and correlating CTI reports, we performed an evaluation study to answer our research question. We performed various IoV attacks that target CAVs in various environments (simulated, real-world, and hybrid). Next, we discuss the setup and the results of the study.

4.1. Setup

This section discusses the selected attack scenarios, the environments, the analysis tools, and the study procedure.

4.1.1. Attack Scenarios

We focused on seven attack scenarios, which are critical for CAVs and IoV systems: GNSS signal spoofing, GNSS signals jamming, CAN DoS attack, alteration of CAN messages, alteration of camera stream, false sensor reading, and malware/ransomware attack. Table 2 presents a brief description of them, and Figure 3 illustrates the IoV environment and the compromised components for each scenario.

Table 2. The attack scenarios performed in our study.

Attack	Description
GNSS signals spoofing	The attacker emits a fake GNSS signal (i.e., providing false positioning and timing data) to the area that the CAV is moving through. The CAV receives this signal and processes false vehicle location. As a result, the moving CAV gets disoriented, demonstrating unexpected and dangerous behavior.
GNSS signals jamming	The attacker attempts to disrupt GNSS communications by jamming GNSS signals. As GNSS signals are vulnerable to intentional interference, the attacker uses a specific device (e.g., Software Defined Radio (SDR) hardware) to overpower GNSS signals so that they cannot be acquired and tracked by the GNSS receiver.
CAN DoS attack	The attacker floods the CAN bus (i.e., the control area network of the CAV that implements a message-based protocol, which enables communication between the in-vehicle components, such as micro-controllers and devices) with messages with high priority. As a result, the CAN bus becomes unresponsive, and thus, there is no control over the CAV, which demonstrates unexpected and dangerous behavior.
Alteration of CAN messages	The attacker tries to manipulate the CAN frames. Data manipulation can be defined as the insertion of an unauthorized CAN frame into the network. Since the CAN protocol does not have an authorization mechanism, a malicious node can attach to the network and inject malicious data. An attacker exploits this vulnerability and sends a CAN message to activate emergency brakes while the vehicle is moving.
Alteration of camera stream	The attacker attempts to alter the video transmitted over the Ethernet network of the CAV. They intercept the video stream and tamper with it. Next, they send it to the supervision center that monitors the CAVs. This results in image elements not being identified by the supervision center correctly. As a result, the supervision center cannot identify the actual situation of the CAV through the internal camera.
False sensor reading	The attacker manipulates the speedometer (i.e., the gauge that measures the instantaneous speed of the CAV) readings to trick the CAV into making bad decisions. The attacker replaces the existing speedometer with a malicious one. Therefore, the control module receives an erroneous value on the speed of the new device, which leads the CAV to make bad decisions and demonstrate unexpected and dangerous behavior.
Malware/Ransomware	The attacker attempts to penetrate and infect the vehicle with malware (e.g., ransomware). They use a Trojan disguised as a legitimate file to trick the end-users into downloading or installing it on a vehicle. The malware encrypts the vehicle's files, making them inaccessible and the vehicle inoperable, and demands a ransom payment to decrypt them.

4.1.2. Environments

- **Simulated environment**. To simulate the attacks on CAVs, we used the co-simulation tool of the nIoVe framework (Figure 4). The co-simulation tool is based on the CARLA open-source simulator that supports the development, training, and validation of autonomous driving systems. The interface of the co-simulation tool (i) enables the security expert to execute a specific attack scenario, (ii) provides a console to view the progress of the attack, and (iii) provides a live video stream of the CAV moving in a metropolitan area.
- **Real-world**. This environment is a real smart city environment. The choice of this scenario is motivated by the fact that a real-life trial presents no danger for humans in this case (e.g., vehicle passengers, pedestrians, and other road users). It is mainly based on the communication between the road-side unit (RSU) and the CAV, with two on-board units (OBUs) installed (one responsible for the vehicle movement and the other responsibilities of the communication).
- **Hybrid**. The deployment and the testing of the cybersecurity framework take part in a real-world-like environment of a setup that meets specific requirements. Like the real-work environment, it is based on the OBUs of the CAV.

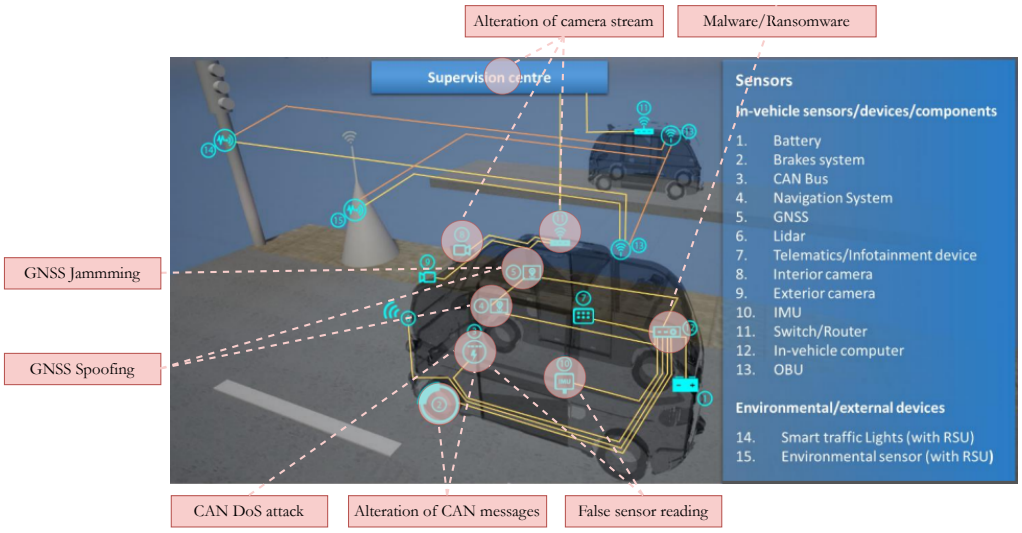


Figure 3. The IoV environment and the compromised components for each attack scenario.

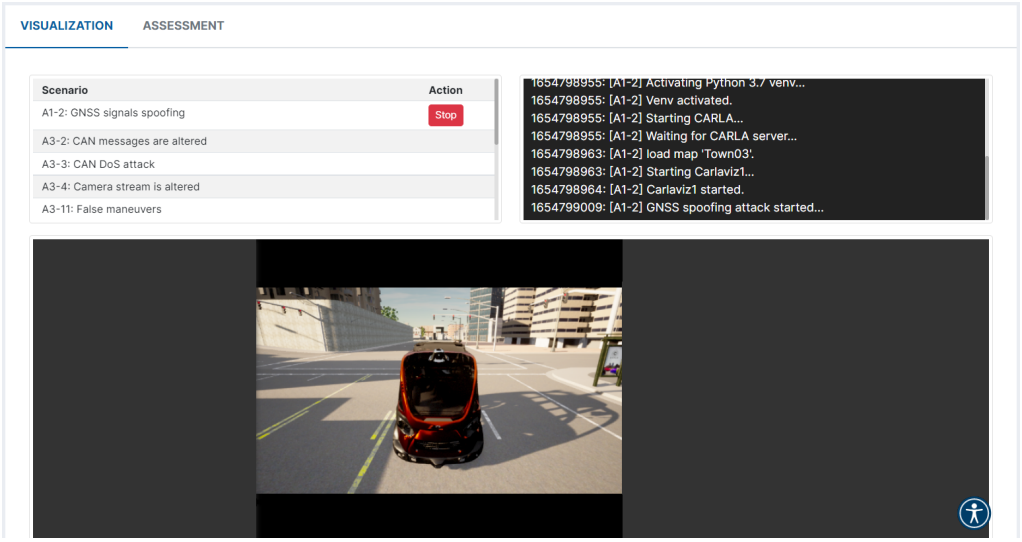


Figure 4. The user interface of the co-simulation tool used for the simulation of the attacks.

4.1.3. Analysis Tools

For the matching and correlation process, we followed a real-time clustering analysis approach using the `scikit-learn` machine-learning library for Python [40]. `scikit-learn` supports several different methods for the clustering of unlabeled data. Given that the AAFRT CTI repository is dynamic and multi-layered, with no predefined clusters of related CTI reports, we focused only on unsupervised machine learning methods, which do not require the pre-specification of the number of clusters. Hence, we took into consideration the `mean-shift`, `DBSCAN`, `OPTICS`, and `BIRCH` methods. Taking into account the increased number of expected clusters (based on the performed attack scenarios) and the increased sample size, we focused on the `DBSCAN` clustering model [41], as discussed in Raptis et al. [39]. The `DBSCAN` model (Appendix A) uses a minimum density level estimation based on a threshold score for the number of neighbors within a specific distance (i.e., radius). Objects with more than that threshold number of neighbors within the area defined by the given radius are considered to form the core clustering points. The `DBSCAN` model aims to find the areas that satisfy the minimum density and which are separated by areas of lower density.

Regarding the features taken into account for the clustering analysis, they included the output of the risk assessment engine of the nIoVe framework, the CAV components that seemed to be influenced during or after the detected malicious activity (e.g., have a different behavior), and potential targets (e.g., CAV components with a high risk of attack propagation). The output of the risk assessment engine provided information about the impact of the malicious activity, such as severity score (i.e., how severe the activity was), probability (i.e., how probable is an ongoing risk occurrence), accuracy (i.e., how precise and accurate the detection was), and vulnerability score (i.e., how the CAV system is influenced by the vulnerability). They are all represented as decimal numbers ranging from 0 to 10.

To assess the clustering performance, we did not use a single metric, but we decided to use a combination of them. In particular, we used the Silhouette coefficient [42], the Calinski–Harabasz score [43], and the Davies–Bouldin index [44]. They are common and well-established metrics that have been widely used for such evaluation processes in the literature. We discuss them next:

- **Silhouette coefficient** [42]. The Silhouette coefficient $s(i)$ is defined for each sample and is composed of two scores, $a(i)$ and $b(i)$. A higher Silhouette coefficient score relates to a model with better defined clusters.

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

$$b(i) = min \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & a(i) < b(i) \\ 0, & a(i) = b(i) \\ b(i)/a(i) - 1, & a(i) > b(i) \end{cases}$$

- **Calinski–Harabasz score** [43]. The Calinski–Harabasz score (also known as the Variance Ratio Criterion) is the ratio of the sum of between-cluster dispersion and of inter-cluster dispersion for all clusters. For a set of data E of size n_E, which has been clustered into k clusters, the Calinski–Harabasz score s is defined as the ratio of the between-cluster dispersion mean and the within-cluster dispersion:

$$s = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k - 1}$$

where $tr(B_k)$ is the trace of the between-cluster dispersion matrix and $tr(W_k)$ is the trace of the within-cluster dispersion matrix defined by:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

with C_q as the set of points in cluster q, c_q the center of cluster q, c_E the center of E, and n_q the number of points in cluster q. The score is higher when the clusters are dense and well separated, which relates to a standard concept of a cluster. Moreover, the score is computed fast.

- **Davies–Bouldin index** [44]. The Davies–Bouldin index signifies the average 'similarity' between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves.

The index is defined as the average similarity between each cluster C_i for $i = 1, ..., k$ and its most similar one C_j. In the context of this index, the similarity is defined as a measure R_{ij} that trades off:

- s_i: the average distance between each point of cluster i and the centroid of that cluster (also known as cluster diameter).
- d_{ij}: the distance between cluster centroids i and j.

A simple choice to construct so that it is nonnegative and symmetric is:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

Then the Davies–Bouldin index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} max(R_{ij})$$

A lower Davies–Bouldin index relates to a model with better separation between the clusters. Zero is the lowest possible score. Values closer to zero indicate a better partition.

4.1.4. Study Procedure

Several attack scenarios have been executed and recorded during the nIoVe EU project. We kept logs of the recorded scenarios and created a dataset with the identified threats along with several features (e.g., computed vulnerability score, potential targets of the system). Our dataset consists of the logs, the assets, and the features collected and calculated during the execution of 2996 attack scenarios.

4.2. Results

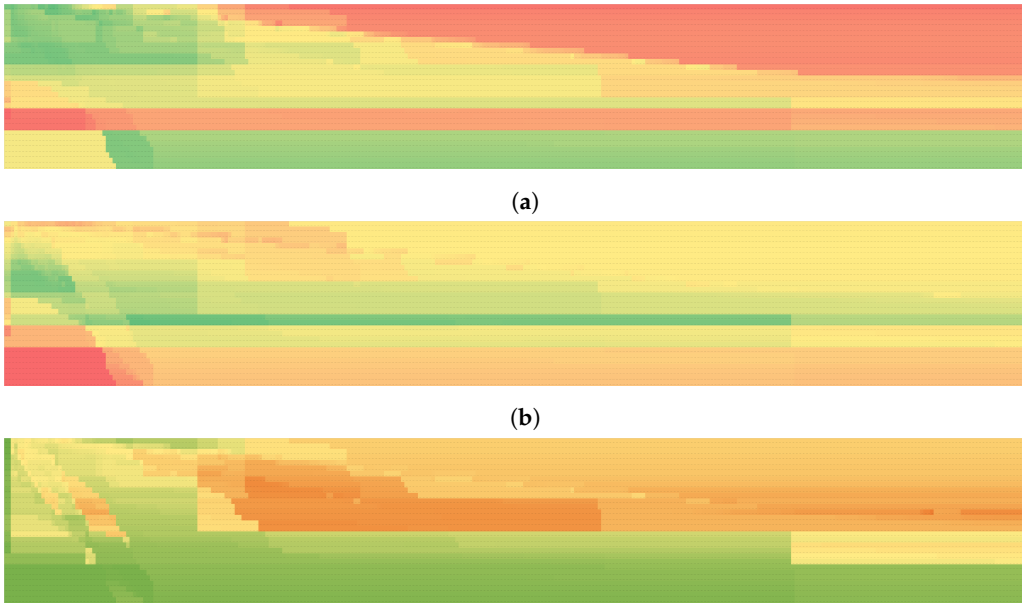
We ran a series of `DBSCAN` tests using the `scikit-learn` machine-learning library with a diverse range of parameters:

- $eps \in [0.1, 3.0]$ with $step = 0.1$, which defines the maximum distance between two samples for one to be considered as in the neighborhood of the other.
- $min_samples \in [1, 300]$ with $step = 1$, which defines the number of samples (i.e., total weight) in a neighborhood for a point to be considered as a core point (including the point itself).

Regarding the analysis of the evaluation metrics, we produced the heatmaps depicted in Figure 5. Each heatmap reflects the assessment of each evaluation metric in a 30×300 matrix. Each row ($N = 30$) represents the eps parameter, and each column ($N = 300$) represents the $min_samples$ parameter. The coloring follows a three-color scheme, with red indicating the poorest performance, yellow indicating the medium performance, and green indicating the best performance. Focusing on the evaluation parameters, this means that the greener a cell is, the higher the Silhouette coefficient is (i.e., close to 1), the higher the Calinski–Harabasz score is, and the lower the Davies–Bouldin index is (i.e., close to 0).

Combining all, we achieved the best results for $eps = 1.1$ and $min_samples = 19$, which provided $N = 7$ core clusters. We should note that the values of the selected parameters are close to the heuristic recommendations suggested by Schubert et al. [45]. Focusing on each metric, we had $s = 0.792$ as the Silhouette coefficient, which is close to the maximum value (i.e., 1), meaning that the core clusters are characterized by high density, there are no overlapping core clusters, they are well apart from each other, and they are clearly distinguished. This is also confirmed by the Calinski–Harabasz score, which indicates, with a maximum value of $s = 2350$, that the core clusters are dense and well separated, relating to a standard concept of a cluster. We should also mention that there is a high degree of correlation between these two metrics and that our results are in line with results of studies in other domains (e.g., [46]), which indicates a high volume of consistency and increased validity. Regarding the Davies–Bouldin index (minimum value: $DB = 0.348$), the analysis

indicates that the clustering model performed well. This is attributed to the fact that the clusters are not similar to each other, and thus, the best clustering scheme minimizes the Davies–Bouldin index.



(**a**)

(**b**)

(**c**)

Figure 5. Heatmaps produced for the evaluation metrics of each experiment run. (**a**) Silhouette coefficient. (**b**) Calinski–Harabasz score. (**c**) Davies–Bouldin index.

We should highlight that the evaluation of CAVeCTIR is based on several parameters, including the `DBSCAN` algorithm, the Silhouette coefficient, the Calinski–Harabasz score, and the Davies–Bouldin index. They all have advantages and disadvantages. `DBSCAN` performs well for datasets with noise, recognizes outliers easily, and the clusters can develop irregular shapes. However, it is sensitive to the *eps* and $min_samples$ parameters, which justifies our decision to explore a variety of combinations. The Silhouette coefficient provides an easy metric for defining the quality of the clustering, but it normally needs high computational effort. The Calinski–Harabasz score tends to be most suitable when the identified clusters are more or less spherical and compact in their middle (e.g., normal distribution). It also tends to have high scores when the clusters have similar sizes. Regarding the Davies–Bouldin index, it has simpler and quicker computation than the Silhouette coefficient, and it is computed based on quantities and features inherent to the available data. However, it is generally higher for convex clusters, it limits the distance metric to the Euclidean space, and a good index value does not always guarantee the best results regarding information retrieval. For all of the above, we decided to use these metrics as complementary to take the best from their application in a real-life IoV scenario.

In an attempt to compare the results derived from our evaluation study with the results derived from other studies conducted in the cybersecurity domain reporting CTI processes, regarding each metric, we observe that:

- **Silhouette coefficient**: our result (s = 0.792) is similar to or better than the coefficient scores reported regarding the clustering of the Conti and Ryuk families (s = 0.725) and Babuk family (s = 0.786) used in CTI processes [47], clusters containing malicious URLs (s = 0.383) [48], and TTP profile extraction and group clustering in IoT (0.610) [49].
- **Calinski–Harabasz score**: our result (s = 2350) is in between the scores reported in other studies regarding TTP profile extraction and group clustering in IoT (s = 3416) [49] and uncovering cybercrimes in social media through natural language processing techniques (s = 1088) [50].

- **Davies–Bouldin index**: our result (DB = 0.348) is similar or lower to the scores reported regarding TTP profile extraction and group clustering in IoT (DB = 0.737) [49] and the enhancement of data quality in real-time CTI (DB = 0.561) [51].
- **Performance timing**: the maximum execution time (in seconds) for the clustering was $t = 209$. The `DBSCAN` performance timings vary widely in the literature (e.g., $t = 11$ [52], $t = 83$ [53], and $t = 2460$ [54]). However, this metric highly depends on the computational resources, the data size, and the complexity of the machine-learning problem. Since CAVeCTIR contributes to the post-incident forensics investigation, the reported timing is acceptable.

5. Discussion

5.1. Summary of Findings

CAVeCTIR is a novel approach that expands the functionality of the nIoVe framework by allowing the automated matching and clustering of new malicious activities with a set of existing CTI reports recorded in an IoV ecosystem. The proposed approach is based on machine-learning techniques and models that enable cybersecurity experts to perform deeper and more accurate post-incident analyses. We also reported an evaluation study in which the proposed approach was activated after a series of different attacks (GNSS signal spoofing, GNSS signals jamming, CAN DoS, alteration of CAN messages, altered camera stream, false sensor reading, and malware attack) targeting the CAVs of an IoV ecosystem. The results provided evidence of the functionality and performance of the proposed approach. Hence, the results lead to active consideration of matching and clustering techniques to automatically find similar CTI reports in IoV environments. Approaches such as CAVeCTIR equip the cybersecurity stakeholders (e.g., organizations, and security experts) with automated tools that find and interrelate similar malicious activities detected in IoV environments quickly with minimum—or even with no—human intervention. As a result, the proposed approach could speed up the post-incident investigation of digital forensics and lead to a more accurate, more efficient, and faster attack attribution.

Attack attribution is a time-consuming, resource-demanding, and challenging task for cybersecurity stakeholders (e.g., organizations, and experts), considering that there are no automated processes for defining and assessing the responsibility for malicious activities, both regarding technical aspects and threat actor profiling. A high volume of information related to such malicious activities, such as IOCs, is available for existing (e.g., previously reported) threat incidents. Cybersecurity stakeholders (e.g., cybersecurity experts and analysts) could benefit from the proposed approach, as they could quickly and accurately interrelate a new malicious activity (e.g., newly detected cyber-attack) with one or more that have already been reported in the CTI repository resulting in a successful attack attribution. We should stress that CAVeCTIR takes into consideration the dynamic, ongoing, and multi-stage nature of the attack attribution process, which often includes malicious activities and incidents of different types and characteristics. During a typical attack attribution scenario, the cybersecurity experts and analysts of an organization would compare information about a new threat to existing knowledge of previous threats recorded in CTI reports. They would evaluate tools, tactics, techniques, and methods of known malicious actors, would assess the collected and analyzed forensics evidence to define a confidence level for their evaluation reports and judgments and would adopt alternative hypotheses and scenarios to trace malicious activities back to their sources, and thus, complete the attack attribution process. In such scenarios, the CAVeCTIR would be a valuable asset for cybersecurity stakeholders to perform an accurate and quick attack attribution.

To provide a clearer picture, let us present an example by making the following hypothesis: an attacker performs a DoS attack on a specific IoV sensor (e.g., GNSS receiver) of a moving CAV. The system detects the anomalies while analyzing the GNSS signals and clusters them into an attack. After the detection of the cyber-attack, the cybersecurity experts of the organization perform a series of post-incident analyses, aiming to attribute the attack (e.g., assign a threat actor profile), among others. The results of the analyses,

performed both by the cybersecurity experts and the IoV system (including both manual and automatic processing ways), constitute the digital forensic evidence, which is part of the recorded CTI report. Forensic evidence includes information such as attack sources, analysis methods, collection tools, tactics, etc. CAVeCTI updates the recorded CTI report whenever a new post-incident analysis is performed or when new evidence is collected and preserved. In our hypothetical scenario, let us also include that the cybersecurity experts of the organization have already evaluated the attack attribution, which resulted in specific threat actors, tools, tactics, techniques, procedures, and incentives. When a new CTI report, which contains information about a detected malicious activity (e.g., GNSS spoofing attack) with similar features and characteristics to old ones, is recorded in the threat intelligence repository, the cybersecurity experts will receive information about the already recorded similar CTI report(s) that have already been processed and assessed. Therefore, cybersecurity experts could adopt similar methods and protocols to collect, store, preserve, process, and analyze the associated forensic evidence, and thus, speed up the attack attribution process eventually. Even in the scenario that the attack attribution process has only partially been completed, CAVeCTIR would enable the cybersecurity experts to identify common patterns and reported characteristics between different CTI reports, weigh their confidence level, and decide the degree of their interrelation, aiming to trace malicious activities back to their origins.

5.2. Integration with Other CTI Repositories

Our approach uses the AAFRT threat intelligence repository to search, collect, and analyze CTI reports. As discussed, the repository is based on the widely used open-source MISP platform. However, our approach is not limited to it. Its open, transparent, dynamic, and adaptive structure allows the cybersecurity stakeholders to expand it and integrate it with other repositories to implement further analysis techniques, including machine learning clustering techniques. For example, the CAVeCTIR could also communicate with the CTI repository of another platform, regardless of its backend system (e.g., MISP platform), to execute all the required operations (e.g., search, collection, and analysis of the recorded CTI reports) to help the cybersecurity stakeholders find related threat incidents quickly while following an automatic process. Another characteristic of our proposed approach is that it can communicate and take benefit of other widely used and established knowledge bases and repositories, such as MITRE ATT&CK (publicly accessible knowledge base of adversary tactics, techniques, and procedures based on real-world malicious incidents and observations) and CAPEC (publicly available catalog of common attack patterns performed by adversaries to exploit known weaknesses in applications and other cyber-enabled capabilities). Toward this direction, the cybersecurity stakeholders could apply diverse formats, protocols, and processes to ensure a high degree of consistency and communication scalability of the shared CTI information (e.g., STIX, TAXII).

5.3. Expandability beyond IoV

The approach we followed focuses on CAVs and IoV environments. However, our approach can be adopted and applied to other environments as well, such as safety-critical systems and e-banking. The only requirement for such environments is to collect, analyze, and record information related to malicious incidents (e.g., cybersecurity threats) to CTI reports. The approach discussed in the manuscript promotes security situational awareness, real-time defense, and in-depth post-incident threat analysis, and thus, it can be applied to diverse domains effectively and efficiently. However, we should stress that the expandability and adaptability of our approach in other than IoV domains might require the analysis and specification of domain-specific characteristics besides the common ones found in diverse types of malicious activities. For example, in the case of an e-banking system, features such as the number of transactions, transferring amounts, routing numbers, and beneficiaries might be critical for the threat analysis. In such cases, the system should adjust and refine the feature models and extraction processes appropriately to support the efficient

matching and correlation of CTI reports in these domains. Therefore, security experts would have the information needed to perform in-depth post-incident forensics analyses and eventually lead to the accurate attribution of the security threat (e.g., incoming attack).

5.4. Limitations and Future Steps

Our approach has three main limitations, which are related to the available data, the parameters of the machine-learning approaches, and the supported malicious activities. The effectiveness and efficiency of the machine-learning approaches (such as the ones used in CAVeCTIR) are associated with the size of the available data and the finetuning of the parameters of the algorithms. CAVeCTIR was based on a sample of almost 3000 malicious incidents covering seven different attack types. Although the size is quite big, the effectiveness of the adopted machine-learning approach and the tuning of its parameters would get better and optimized during the ongoing evolution of the system. Focusing on the machine learning model and its characteristics, CAVeCTIR was limited to the use of the DBSCAN algorithm, considering that our previous work indicated its suitability in such cases. Moreover, we focused on a specific set of features and parameters, which are dependent on the adopted IoV framework (i.e., nIoVe) and the application of the machine learning model. The extraction and selection of more features, along with the re-configuration of the parameters of the adopted machine learning models (DBSCAN and other algorithms), should be explored. Regarding the supported malicious activities, CAVeCTIR is limited to seven main types: GNSS signal spoofing, GNSS signals jamming, CAN DoS attack, alteration of CAN messages, alteration of camera stream, false sensor reading, and malware/ransomware attack. CAVeCTIR could consider more types in the future to support a wider area of malicious activities. These types include: V2X signals jamming and spoofing, jamming of mobile broadband communications signals, leakage of IoV credentials, alteration of remote commands, and alteration of embedded firmware.

The presented work can be extended with our future goals falling into three major categories: (i) optimization of the clustering analysis, (ii) integration with other CTI systems, and (iii) thorough evaluation by security experts. Regarding the optimization of the clustering analysis, we plan to (i) analyze more IoV attacks focusing on CAVs (e.g., alteration of CAN bus messages, false maneuvers, traffic lights alterations, and V2X attacks) in other than simulation environments (e.g., hybrid and real-word environments), and (ii) explore combined clustering methods and diverse threat parameters/features, aiming to refine the models, improve their performance, and thus, provide enhanced assistance to the security experts. Regarding the integration with other CTI systems, we plan to extend the proposed approach to other CTI repositories of IoV environments (e.g., in nIoVe, there is a shared threat intelligence repository that records a complete list of incoming threats, including response and recovery actions). We also plan to extend the proposed approach to communicate with open-source and publicly accessible platforms, including knowledge bases and repositories (e.g., MITRE ATT&CK). This extension would help security experts to perform a more detailed forensics analysis, given that they would have access to more information related to the malicious incident they investigate. Finally, regarding the evaluation by security experts, we plan to perform thorough user evaluation studies in diverse environments (e.g., simulation, hybrid, and real-world), where experienced security stakeholders (e.g., security experts) will assess several dimensions of the proposed approach, such as its functionality, operability, and efficiency.

6. Conclusions

In this manuscript, we presented CAVeCTIR, a novel approach that could support the cybersecurity stakeholders (e.g., experts and organizations) during post-incident forensic analysis. CAVeCTIR seeks, collects, analyzes, and correlates cyber threat intelligence (CTI) reports that record detected malicious activities on connected and autonomous vehicles (CAVs) in Internet-of-Vehicles (IoV) environments. CAVeCTIR takes advantage of machine learning techniques (e.g., real-time clustering analysis) to find similarities between

new and recorded CTI reports. Hence, cybersecurity stakeholders could gain access to previous knowledge, identify common patterns, and adopt similar techniques and tools during the post-incident investigation. Therefore, it could drive a more effective and accurate forensics analysis and, thus, speed up the attack attribution process. CAVeCTIR helps cybersecurity stakeholders to mitigate the impact of the detected malicious activities quickly and effectively, and eventually, prevent future similar activities in IoV environments. While CAVeCTIR focuses on CAVs and IoV, it could also be applied to other application domains, such as Industrial Internet-of-Things (IIoT), e-banking, and safety-critical systems. However, it might need appropriate adjustments and adaptations to fit the unique features and address the challenges of each application domain.

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

Funding: This research was funded by European Union (EU) Horizon 2020 research and innovation programme—nIoVe: A Novel Adaptive Cybersecurity Framework for the Internet-of-Vehicles under grant agreement No 833742.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The co-funding body had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript (main text, tables, figures):

CAV	Connected and Autonomous Vehicle
IoV	Internet of Vehicles
IOCs	indicators of compromise
CAN bus	Controller Area Network bus
GNSS	Global Navigation Satellite System
DoS	Denial of Service
IMU	Inertial Measurement Unit
OBU	On-Board Unit
RSU	Road-Side Unit

Appendix A

The `DBSCAN` algorithm used in the CAVeCTIR model can be expressed in pseudocode as follows:

```
DBSCAN(D, eps, MinPts):
    C = 0
    for each unvisited point P in dataset D:
        mark P as visited
        NeighborPts = regionQuery(P, eps)
        if sizeof(NeighborPts) < MinPts:
            mark P as NOISE
        else:
            C = next cluster
```

```
                    expandCluster(P, NeighborPts, C, eps, MinPts)


expandCluster(P, NeighborPts, C, eps, MinPts):
    add P to cluster C
    for each point P' in NeighborPts:
        if P' is not visited:
            mark P' as visited
            NeighborPts' = regionQuery(P', eps)
            if sizeof(NeighborPts') >= MinPts:
                NeighborPts = NeighborPts joined with NeighborPts'
        if P' is not yet member of any cluster:
            add P' to cluster C


regionQuery(P, eps):
    return all points within P's eps-neighborhood (including P)
```

References

1. Sun, Y.; Wu, L.; Wu, S.; Li, S.; Zhang, T.; Zhang, L.; Xu, J.; Xiong, Y.; Cui, X. Attacks and Countermeasures in the Internet of Vehicles. *Ann. Telecommun.* **2016**, *72*, 283–295. [CrossRef]
2. Kim, K.; Kim, J.S.; Jeong, S.; Park, J.H.; Kim, H.K. Cybersecurity for Autonomous Vehicles: Review of Attacks and Defense. *Comput. Secur.* **2021**, *103*, 102–150. [CrossRef]
3. Serinelli, B.M.; Collen, A.; Nijdam, N.A. Training Guidance with KDD Cup 1999 and NSL-KDD Data Sets of ANIDINR: Anomaly-Based Network Intrusion Detection System. *Procedia Comput. Sci.* **2020**, *175*, 560–565. [CrossRef]
4. Hamad, M.; Tsantekidis, M.; Prevelakis, V. Intrusion Response System for Vehicles: Challenges and Vision. In *Communications in Computer and Information Science*; Springer International Publishing: Cham, Switzerland, 2021; pp. 321–341. [CrossRef]
5. Noor, U.; Anwar, Z.; Amjad, T.; Choo, K.K.R. A Machine Learning-based FinTech Cyber Threat Attribution Framework using High-level Indicators of Compromise. *Future Gener. Comput. Syst.* **2019**, *96*, 227–242. [CrossRef]
6. Gao, Y.; Xiaoyong, L.; Hao, P.; Fang, B.; Yu, P. HinCTI: A Cyber Threat Intelligence Modeling and Identification System Based on Heterogeneous Information Network. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 708–722. [CrossRef]
7. Preuveneers, D.; Joosen, W. Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence. *J. Cybersecur. Priv.* **2021**, *1*, 140–163. [CrossRef]
8. Koloveas, P.; Chantzios, T.; Alevizopoulou, S.; Skiadopoulos, S.; Tryfonopoulos, C. inTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence. *Electronics* **2021**, *10*, 818. [CrossRef]
9. Zacharaki, A.; Paliokas, I.; Votis, K.; Alexakos, C.; Serpanos, D.; Tzovaras, D. Complex Engineering Systems as an Enabler for Security in Internet of Vehicles: The nIoVe Approach. In Proceedings of the 2019 First International Conference on Societal Automation (SA), Krakow, Poland, 4–6 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8. [CrossRef]
10. Conti, M.; Dargahi, T.; Dehghantanha, A. Cyber Threat Intelligence: Challenges and Opportunities. In *Advances in Information Security*; Springer International Publishing: Cham, Switzerland, 2018; pp. 1–6. [CrossRef]
11. Mavroeidis, V.; Bromander, S. Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
12. Barnum, S. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). *Mitre Corp.* **2012**, *11*, 1–22.
13. Wagner, T.D.; Palomar, E.; Mahbub, K.; Abdallah, A.E. A Novel Trust Taxonomy for Shared Cyber Threat Intelligence. *Secur. Commun. Netw.* **2018**, *2018*, 9634507. [CrossRef]
14. Schlette, D.; Böhm, F.; Caselli, M.; Pernul, G. Measuring and visualizing cyber threat intelligence quality. *Int. J. Inf. Secur.* **2020**, *20*, 21–38. [CrossRef]
15. Landauer, M.; Skopik, F.; Wurzenberger, M.; Hotwagner, W.; Rauber, A. A Framework for Cyber Threat Intelligence Extraction from Raw Log Data. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; IEEE: Piscataway, NJ, USA, 2019. [CrossRef]
16. Zhao, J.; Yan, Q.; Li, J.; Shao, M.; He, Z.; Li, B. TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data. *Comput. Secur.* **2020**, *95*, 101867. [CrossRef]
17. Deliu, I.; Leichter, C.; Franke, K. Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]

18. Kadoguchi, M.; Hayashi, S.; Hashimoto, M.; Otsuka, A. Exploring the Dark Web for Cyber Threat Intelligence using Machine Leaning. In Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 1–3 July 2019; IEEE: Piscataway, NJ, USA, 2019. [CrossRef]
19. Zhang, H.; Shen, G.; Guo, C.; Cui, Y.; Jiang, C. EX-Action: Automatically Extracting Threat Actions from Cyber Threat Intelligence Report Based on Multimodal Learning. *Secur. Commun. Netw.* **2021**, *2021*, 5586335. [CrossRef]
20. Li, K.; Wen, H.; Li, H.; Zhu, H.; Sun, L. Security OSIF: Toward Automatic Discovery and Analysis of Event Based Cyber Threat Intelligence. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
21. Zhu, Z.; Dumitras, T. ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
22. Husari, G.; Niu, X.; Chu, B.; Al-Shaer, E. Using Entropy and Mutual Information to Extract Threat Actions from Cyber Threat Intelligence. In Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), Miami, FL, USA, 9–11 November 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
23. Ghazi, Y.; Anwar, Z.; Mumtaz, R.; Saleem, S.; Tahir, A. A Supervised Machine Learning Based Approach for Automatically Extracting High-Level Threat Intelligence from Unstructured Sources. In Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 17–19 December 2018; IEEE: Piscataway, NJ, USA, 2018. [CrossRef]
24. Wang, X.; Chen, R.; Song, B.; Yang, J.; Jiang, Z.; Zhang, X.; Li, X.; Ao, S. A Method for Extracting Unstructured Threat Intelligence Based on Dictionary Template and Reinforcement Learning. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Piscataway, NJ, USA, 2021. [CrossRef]
25. Wagner, T.D.; Mahbub, K.; Palomar, E.; Abdallah, A.E. Cyber threat intelligence sharing: Survey and research directions. *Comput. Secur.* **2019**, *87*, 101589. [CrossRef]
26. Wagner, C.; Dulaunoy, A.; Wagener, G.; Iklody, A. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, WISCS '16, Vienna, Austria, 24 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 49–56. [CrossRef]
27. Gao, P.; Liu, X.; Choi, E.; Soman, B.; Mishra, C.; Farris, K.; Song, D. A System for Automated Open-Source Threat Intelligence Gathering and Management. In Proceedings of the 2021 International Conference on Management of Data, Online, China, 20–25 June 2021; ACM: New York, NY, USA, 2021. [CrossRef]
28. Connolly, J.; Davidson, M.; Schmidt, C. The Trusted Automated eXchange of Indicator Information (TAXII). *Mitre Corp.* **2014**. Available online: https://taxii.mitre.org/about/documents/Introduction_to_TAXII_White_Paper_May_2014.pdf (acessed on 12 October 2022).
29. Kukkala, V.K.; Thiruloga, S.V.; Pasricha, S. Roadmap for Cybersecurity in Autonomous Vehicles. *IEEE Consum. Electron. Mag.* **2022**, *11*, 13–23. [CrossRef]
30. He, H.; Gray, J.; Cangelosi, A.; Meng, Q.; McGinnity, T.; Mehnen, J. The challenges and opportunities of artificial intelligence in implementing trustworthy robotics and autonomous systems. In Proceedings of the 3rd International Conference on Intelligent Robotic and Control Engineering, Oxford, UK, 10–12 August 2020.
31. Panda, S.; Rass, S.; Moschoyiannis, S.; Liang, K.; Loukas, G.; Panaousis, E. HoneyCar: A Framework to Configure Honeypot Vulnerabilities on the Internet of Vehicles. *arXiv* **2021**, arXiv:2111.02364.
32. Basnet, M.; Ali, M. A Deep Learning Perspective on Connected Automated Vehicle (CAV) Cybersecurity and Threat Intelligence. *arXiv* **2021**, arXiv:2109.10763.
33. Ali, E.S.; Hasan, M.K.; Hassan, R.; Saeed, R.A.; Hassan, M.B.; Islam, S.; Nafi, N.S.; Bevinakoppa, S. Machine learning technologies for secure vehicular communication in internet of vehicles: Recent advances and applications. *Secur. Commun. Netw.* **2021**, *2021*, 8868355. [CrossRef]
34. Liu, W.; Xu, X.; Qi, L.; Zhang, X.; Dou, W. GoDeep: Intelligent IoV Service Deployment and Execution with Privacy Preservation in Cloud-edge Computing. In Proceedings of the 2021 IEEE International Conference on Web Services (ICWS), Chicago, IL, USA, 5–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 579–587.
35. Mohseni, S.; Pitale, M.; Singh, V.; Wang, Z. Practical solutions for machine learning safety in autonomous vehicles. *arXiv* **2019**, arXiv:1912.09630.
36. Kumar, P.; Gupta, G.P.; Tripathi, R.; Garg, S.; Hassan, M.M. DLTIF: Deep Learning-Driven Cyber Threat Intelligence Modeling and Identification Framework in IoT-Enabled Maritime Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
37. Alexakos, C.; Katsini, C.; Votis, K.; Lalas, A.; Tzovaras, D.; Serpanos, D. Enabling Digital Forensics Readiness for Internet of Vehicles. *Transp. Res. Procedia* **2021**, *52*, 339–346. [CrossRef]
38. Katsini, C.; Raptis, G.E.; Alexakos, C.; Serpanos, D. FoRePlan: Supporting Digital Forensics Readiness Planning for Internet of Vehicles. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, PCI 2021, Volos, Greece, 26–28 November 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 369–374. [CrossRef]

39. Raptis, G.E.; Katsini, C.; Alexakos, C. Towards Automated Matching of Cyber Threat Intelligence Reports based on Cluster Analysis in an Internet-of-Vehicles Environment. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 26–28 July 2021; IEEE: Piscataway, NJ, USA, 2021. [CrossRef]
40. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
41. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Kdd, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
42. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
43. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.-Theory Methods* **1974**, *3*, 1. [CrossRef]
44. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227.
45. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.* **2017**, *42*, 19:1–19:21. [CrossRef]
46. Wang, X.; Xu, Y. An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; Volume 569, p. 052024.
47. Grelot, F.; Larinier, S.; Salmon, M. Automation of Binary Analysis: From Open Source Collection to Threat Intelligence. In Proceedings of the 28th C&ESAR, Rennes, France, 16–17 November 2021; CEUR-WS: Aachen, Germany, 2021; pp. 41–56.
48. Nayak, S.; Nadig, D.; Ramamurthy, B. Analyzing Malicious URLs using a Threat Intelligence System. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.
49. Wu, Y.; Huang, C.; Zhang, X.; Zhou, H. GroupTracer: Automatic attacker TTP profile extraction and group cluster in Internet of things. *Secur. Commun. Netw.* **2020**, *2020*, 8842539. [CrossRef]
50. Ramírez Sánchez, J.; Campo-Archbold, A.; Zapata Rozo, A.; Díaz-López, D.; Pastor-Galindo, J.; Gómez Mármol, F.; Aponte Díaz, J. Uncovering cybercrimes in social media through natural language processing. *Complexity* **2021**, *2021*, 7955637. [CrossRef]
51. Rodriguez, A.; Okamura, K. Enhancing data quality in real-time threat intelligence systems using machine learning. *Soc. Netw. Anal. Min.* **2020**, *10*, 91. [CrossRef]
52. Smiti, A.; Elouedi, Z. Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques. In Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), Lisbon, Portugal, 13–15 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 573–578.
53. Mustakim; Fauzi, M.Z.; Mustafa; Abdullah, A.; Rohayati. Clustering of Public Opinion on Natural Disasters in Indonesia Using DBSCAN and K-Medoids Algorithms. *J. Phys. Conf. Ser.* **2021**, *1783*, 012016. [CrossRef]
54. Sarma, A.; Goyal, P.; Kumari, S.; Wani, A.; Challa, J.S.; Islam, S.; Goyal, N. μDBSCAN: An exact scalable DBSCAN algorithm for big data exploiting spatial locality. In Proceedings of the 2019 IEEE International Conference on Cluster Computing (CLUSTER), Albuquerque, NM, USA, 23–26 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–11.