IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 December 2024, accepted 28 December 2024, date of publication 23 January 2025, date of current version 3 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3533084

RESEARCH ARTICLE

AI-Based Holistic Framework for Cyber Threat Intelligence Management

ARNOLNT SPYROS, ILIAS KORITSAS, ANGELOS PAPOUTSIS, PANOS PANAGIOTOU, DESPOINA CHATZAKOU, DIMITRIOS KAVALLIEROS, THEODORA TSIKRIKA, STEFANOS VROCHIDIS, (Member, IEEE), AND IOANNIS KOMPATSIARIS, (Senior Member, IEEE)

Information Technologies Institute, Centre for Research and Technology Hellas, 570 01 Thessaloniki, Greece

Corresponding author: Arnolnt Spyros (aspyros@iti.gr)

This work was supported in part by European Union's Horizon 2020 Research and Innovation Program under Grant 101021727 (IRIS) and Grant 101019645 (SECANT), and in part by the Horizon Europe Research and Innovation Program under Grant 101070670 (ENCRYPT).

ABSTRACT Cyber Threat Intelligence (CTI) is an important asset for organisations to facilitate the safeguarding of their systems against new and emerging cyber threats. CTI continuously provides up-to-date information which enables the design and implementation of better security measures and mitigation strategies. Organisations gather data from different sources either internal or external to the organisation, which are analysed, resulting in CTI. Nevertheless, the gathered data usually contain a large amount of content that is irrelevant to CTI or even to cybersecurity. Furthermore, most approaches concerning CTI management (e.g., gathering, analysis) involve simply gathering and storing the information without any enrichment such as classification or correlation. However, in order to obtain optimal results, organisations should be able to utilise all capabilities of CTI. Therefore, in this work, we propose ThreatWise AI, a novel framework that enables the gathering, analysis, enrichment, storage, and sharing of CTI in an efficient and secure manner. In particular, we have developed a novel pipeline in ThreatWise AI which incorporates different advanced tools, with distinct capabilities that interact with each other to provide a complete set of functionalities for the administration of the overall CTI lifecycle. The developed tools integrate various Python scripts and provide gathering and analysis functionalities of CTI. Furthermore, the proposed framework leverages the MISP platform for storing, enriching and sharing while also integrating Artificial Intelligence (AI) and Machine Learning (ML) algorithms for advanced data enrichment.

INDEX TERMS Artificial intelligence, cyber threat intelligence, data classification, data correlation, honeypots, machine learning, named entity recognition, outlier detection, social media crawling, web crawling.

I. INTRODUCTION

cyber-attacks are continuously increasing and becoming more sophisticated, while also requiring less effort and technical knowledge by those behind them [1], [2], [3]. Many organisations are susceptible to both known, as well as zero-day (unknown) vulnerabilities, with the majority of attacks targeting critical infrastructures [4]. Successful compromise of such infrastructures could have serious consequences by affecting critical operations, such as monitoring the temperature of thermal reactors, which could even lead to physical damages and considerable business impact. Apart from the physical damages and business impact, such cybersecurity incidents could also result in legal liabilities for the affected organisations (e.g., GDPR sanctions). Consequently, organisations need to implement appropriate security mechanisms to enhance the cybersecurity of their infrastructures against the various existing and especially emerging threats.

To effectively address such continuously evolving attacks, it is important to obtain intelligence regarding the threat

The associate editor coordinating the review of this manuscript and approving it for publication was Bang L. H. Nguyen.

landscape, and more specifically intelligence concerning the severity and nature of the attacks. In this regard, more and more organisations implement the concept of Cyber Threat Intelligence (CTI), which includes intelligence in a structured manner concerning cyber threats and vulnerabilities, obtained after the relevant information has been collected, aggregated, evaluated, analysed, or enriched using appropriate techniques [5]. Essentially, CTI provides relevant, up-to-date and actionable information about existing, new and emerging threats within the threat landscape, enabling organisations to identify, assess, monitor, and respond to cyber threats targeting their systems and infrastructures.

CTI also constitutes a valuable source towards raising cybersecurity awareness by being leveraged for training the organisations' staff against current and emerging threats. In particular, CTI could provide adequate training material since it mostly includes technical information such as IPv4 addresses, file hashes, and URLs known as Indicators of Compromise (IoCs). Hence, CTI can also facilitate the creation of realistic training scenarios for different threats and different industry domains (e.g., Aviation, Naval, Power Grid, Smart Cities, and Healthcare ecosystems).

A. CTI GATHERING

The gathering of CTI initially requires the identification of the CTI sources. CTI sources and the threat information that needs to be collected from monitoring devices which will facilitate decision-making processes, are defined within this step. Subsequently, CTI is gathered from the identified sources in accordance with a defined procedure. The gathering might include the use of external (i.e., online) as well as internal, to the organisation, sources for extracting a wide variety of information [6], [7], [8]. *External sources* include sources such as CERT and CSIRT feeds, malware repositories, X (former Twitter) and other relevant feeds. On the other hand, *internal sources* are defined as sources that are internal to the organisation and include logs generated by servers, logs from databases, security monitoring tools (e.g., IDS, IPS), and various other services which operate within the organisation.

More in particular, data gathering from external sources can include both high-level (e.g., articles concerning cyber threats) as well as technical information, including IoCs, Indicators of Attack (IoAs) and Tactics, Techniques, and Procedures (TTP). Similarly, internal sources are very useful for collecting intelligence regarding threats and zero-day vulnerabilities. Zero-day vulnerabilities can be detected through anomalies that are identified within the internal networks of organisations. Moreover, various sources (e.g., system logs, database logs, etc.) can be combined to chart the entire behaviour of the attackers, including the TTPs they utilise. Therefore, the gathered data could result in heterogeneous information, including information regarding the attacker's tactics and procedures and also technical information such as IoCs.

However, despite the various advantages concerning internal sources, there are also specific challenges that might emerge when leveraging them for CTI data extraction, which should be considered. First, the automated analysis of logs could be challenging in terms of interpreting the data and extracting exploitable information. Commonly, logs from internal sources use human-readable syntax which obfuscates the automatic log parsing [8]. Furthermore, log formats could be subject to changes over time, and thus log parsers should be adapted each time according to the new format.

B. CTI SHARING AND ANALYSIS

With regard to the sharing of CTI information between organisations, it enhances the knowledge, experience, and prevention capabilities of each participating organisation against previously identified or emerging cyber threats. CTI sharing facilitates the joint effort towards the defence against cyber-attacks since more organisations are able to collect CTI and, in some cases enrich the data. Furthermore, the security posture of any organisation which participates in such a collaboration scheme is enhanced, since their CSIRTs are able to plan and develop the necessary countermeasures for the timely detection of the latest kind of attacks.

In this regard, there is a need for efficient and automated tools for analysing and sharing heterogeneous CTI related to the present systems' configurations, attacker's threats and tactics, and indicators of ongoing incidents, to build proper and effective defensive capabilities. Faced with the numerous architectures, products, and systems being used as sources of data for information-sharing systems, there is a need for standardised and structured CTI platforms to allow a satisfying level of interoperability across the various stakeholders.

In this context, this work proposes ThreatWise AI, a novel holistic approach towards the gathering, analysis, enrichment, storing, and sharing of CTI data. Specifically, the ThreatWise AI introduces a novel framework which integrates different novel components. The developed web and social media crawlers and the implemented Wazuh instances, enable the collection, extraction, and enrichment of CTI, both from external and internal sources. Internal sources include, among others, logs generated by servers and databases, security monitoring tools (e.g., Intrusion Detection System (IDS) tools, Intrusion Prevention System (IPS) tools), honeypot instances, and other services that are deployed and operate within an organisation. External sources, on the other hand, include sources that are located outside the organisation's premises, such as CERT and CSIRT feeds, vulnerability databases, social media platforms, and other relevant feeds. Besides, the IS component enables the storing, correlation (i.e., enrichment), and sharing of the extracted CTI in a secure and efficient manner.

The collection process is achieved both (i) manually through a user-friendly Graphical User Interface (GUI) as well as (ii) automatically on a daily basis from trusted sources, leveraging appropriate scripts and configurations. The collected information from all sources is initially filtered to avoid storing Personal Identifiable Information (PII), by leveraging rule-based techniques, whereas then CTI is extracted using rule-based and Machine Learning (ML) based techniques. Subsequently, the extracted CTI is further analysed and enriched by leveraging correlation and data classification techniques. Possible correlations between the information are identified by implementing both simple and advanced correlation techniques. More specifically, simple correlation concerns the identification of similar values in different fields, such as the same source IP address from where the attack has originated. On the other hand, advanced correlation performs correlation according to different features that are extracted from the identified threats. Simple correlation uses the default correlation engine of MISP - Open-source Threat Intelligence & Open standards for threat information sharing,[1] whereas advanced correlation leverages advanced ML and Artificial Intelligence (AI) algorithms.

The functionalities provided by the developed tools of ThreatWise AI, allow for an organisation to remain updated on the current as well as emerging cyber threats. The users are able to gather, utilise and share CTI in a secure and efficient manner by implementing authentication and authorisation mechanisms either via the user interface or an Application Protocol Interface (API). ThreatWise AI integrates novel tools which enable the enrichment of the extracted CTI, thus allowing the creation of more complex rules in terms of prevention, identification and mitigation of cyber threats within the infrastructure of the organisation.

C. CONTRIBUTIONS

Overall, the main contributions of the proposed approach are:

- The ThreatWise AI constitutes an innovative framework concerning gathering, analysing, enriching and sharing CTI in an efficient, secure, and user-friendly manner.
- The proposed framework enables data gathering from heterogeneous online sources, including social media platforms (i.e., X and Reddit).
- The proposed framework includes advanced AI and ML algorithms for threat correlation.
- Data classification leveraging advanced AI and ML algorithms.
- Increased actionability of shared CTI.

The rest of the paper is organised as follows: Section II presents the background and related work. Section III describes the contribution of this paper. The proposed framework is conceptually explained in Section IV, whereas Section V delves into details regarding the technical development of the components. The conducted experiments are presented in Section VI. Section VII explains the evaluation metrics that have been leveraged during the experiments. In Section VIII we discuss the key findings of the proposed approach while providing a summary of the paper and future research directions. Finally, Section IX concludes this study including future improvements and enhancements of the proposed approach.

II. BACKGROUND & RELATED WORK

A. INFORMATION GATHERING

To address the constantly increasing cybersecurity issues, organisations have to retain visibility of existing, emerging and evolving cyber threats, implement appropriate proactive and reactive measures, as well as to define effective mitigation strategies. CTI is an important asset for an organisation towards the improvement and enhancement of their security capabilities and according to the authors in [9], a proper assessment of the scale of risk within an organisation requires information from internal as well as external sources; internal information can be defined as *Local domain knowledge* that is used to determine the risk associated with internal assets, whereas external information can be defined as *Global domain knowledge* that concerns data collected by external sources and processed internally and is used to augment the *Local domain knowledge*.

Internal assets of an organisation can include system logs, network events, application and cybersecurity incident reports [7]. While they provide valuable data such as IoCs, they also include a large amount of benign traffic. Therefore, whereas the volume of the generated data can be high, the actual valuable information concerning CTI can be very limited.

Another critical internal CTI source is honeypot solutions which are the main source of the proposed framework, in terms of internal sources. Honeypots are defined as decoy systems to attract attackers by exposing vulnerable services. The first mention of the term "honeypot" was by Spitzner [10], who defined them as a "security resource whose value lies in being probed, attacked, or compromised". Overall, honeypots are considered a valuable source concerning the gathering of tactics, techniques, and attack patterns that adversaries use for the exploitation of the exposed services. Their usability and value originate from their effectiveness in terms of prevention, detection, and reaction against cyber-attacks [11]. There is a variety of different types of honeypots which are categorised according to distinct criteria, such as their field of operation and the level of interaction (i.e., as low-, medium-, and high-interaction honeypots [12]).

In recent years, there has been active research on utilising honeypot instances for the collection and extraction of CTI data [13], [14], [15], [16], [17], [18], [19], [20]. Most proposed frameworks deploy either different low or medium-interaction honeypots, with some of them deploying multiple instances of the same honeypot. Concerning the analysis and visualisation of the collected data, either from honeypots or from other sources, some approaches utilise the Elastic Stack (ELK) [16] whereas others use the MISP threat intelligence sharing platform [17], [18].

[1] https://www.misp-project.org

Moreover, honeypots can also operate with other security solutions (e.g., IDS and Security Information and Event Management (SIEM) solutions) to improve and enhance their detection performance [14], [15]. Considering that the information stored in honeypots is mostly technical, it includes many IoCs which could greatly increase the quality of the generated CTI. In particular, the identified IoCs can feed an IPS or SIEM system in order to compose the appropriate rules to detect or even mitigate the cyber-attacks associated with these IoCs.

Concerning the CTI gathering and extraction from honeypots, the majority of approaches in the literature utilise docker containers for the deployment of honeypots. Medium-interaction and low-interaction honeypots are preferred, with Cowrie and Dionaea being the most popular ones [13], [14], [15], [16], [17]. For instance, authors in [16] deploy different honeypots, namely Cowrie,[2] Dionaea,[3] and Whaler,[4] to assess the security of data-connections of a house. The deployment of honeypots has been performed on a Raspberry Pi [21], leveraging Docker[5] containers, enabling a layer of isolation. Following the gathering process, the data is then sent to the ELK[6] stack for analysis and visualisation. A similar approach is followed in [13] where the authors have developed a multi-component honeypot system (i.e., Cowrie, Dionaea, and Glastopf[7]) deployed in Docker containers. The produced logs are gathered and analysed by custom scripts that convert the collected data into a JSON format. Moreover, a cloud-based LAN-security monitoring system based on honeypots is presented in [15]. The system comprises a monitoring node deployed on a Raspberry Pi 3 board, running several honeypot instances (i.e., Cowrie, Dionaea) that interact with malware within the LAN network. The gathered information is sent to a data collection processing server where a custom algorithm recognises malicious events.

High-interaction honeypots have also been leveraged to collect data concerning cybersecurity attacks [17]. Relevant data is gathered and subsequently correlated with the tactics and techniques described in the ATT&CK framework[8] developed by the MITRE Corporation,[9] enabling the identification of known adversary patterns. Despite the fact that high-interaction honeypots are considered the most sophisticated type, they introduce high complexity regarding their deployment and maintenance.

Focusing on external sources, gathering of CTI data can be achieved with the implementation of simple approaches such as REST APIs or more advanced ones such as web crawlers [22], [23]. Essentially, web crawlers visit a target URL address and download the content. Subsequently, the crawler identifies and extracts the hyperlinks and compares them with a list of visited URLs, adding the non-visited ones to its frontier list. The procedure is repeated for all the ranges of the domain or sub-domain until it is fully crawled [23].

Overall, web crawlers are organised in different categories according to the set of features they support, such as crawling application, the available hardware, the desired scalability properties, and the ability to scale/expand the existing infrastructure [24], [25], [26]. In particular, they can be categorised into the following high-level categories:

- **Centralised:** Centralised crawlers can be either special-purpose or small crawlers which are based on a centralised architecture [24].
- **Parallel/Distributed:** This type of crawlers implements multiple crawling processes (referred to as C-processing crawler jargon) that implement all the basic crawling functionalities [27], [28].
- **Hybrid:** The hybrid architecture, which combines centralised and distributed architectures, is considered the predominant architecture since it has a simple design, providing the ability to distribute some of the processes, whereas others can remain centralised [22].
- **Peer-to-peer:** Peer-to-peer architecture refers to a special type of distributed crawlers that are intended to operate on machines located at the edge of the internet [29], [30].

All in all, web crawlers are considered the optimal approach to gather CTI from external sources considering that they enable the gathering from all web layers, namely Surface, Deep and Dark web.

B. CTI ANALYSIS AND ENRICHMENT

While CTI contains valuable information such as information about the attacker and the compromised asset, additional actionable information could be added through enrichment to further increase the value and usefulness of the available CTI. CTI enrichment can be achieved by classifying and correlating the collected information, which can significantly increase the quality and actionability of the gathered CTI.

1) CLASSIFICATION

There are some research efforts towards addressing the problem of *CTI classification*, which refers to classifying a text as containing CTI-related information or not. For instance, a framework for gathering CTI from posts of the X social media platform has been developed in [31], which builds on top of a classifier that is responsible for characterising texts as related to CTI or not, trained on features extracted from public repositories, such as Common Vulnerabilities and Exposures (CVE).

Text classification can also be used to filter the information available to ultimately retain the one that aligns with the domain of interest (i.e. *domain classification*). In the context of our application, we are interested in selecting texts that belong to either the aviation, naval, or power grid sectors, all of which are considered critical cybersecurity sectors.

[2] https://github.com/cowrie/cowrie
[3] https://github.com/DinoTools/dionaea
[4] https://github.com/oncyberblog/whaler
[5] https://www.docker.com/
[6] https://www.elastic.co/elastic-stack
[7] https://github.com/mushorg/glastopf
[8] https://attack.mitre.org
[9] https://www.mitre.org

In contrast to the CTI classification, where the documents could fall into only one of the three classes (either "CTI-related", "cybersecurity-related but with no CTI content", or "not related to cybersecurity"), in domain classification a document may contain content about more than one domain (for example, it could describe a vulnerability that affects both the naval and the aviation sector). This problem setting is known as multi-label classification[10] and appropriate techniques must be used [32].

For an effective text-based classification, an important factor is how the data (i.e., words) are represented (encoded) before being given as input to the corresponding classification algorithm. Encoding words as a machine-readable language model is often performed using different semantic representation methods, such as Word2Vec [33], GloVe [34], and ELMo [35]. Compared to the more traditional methods (such as Word2Vec and GloVe), the more recent method known as ELMo (Embeddings from Language Models) introduces a context-based text representation approach, meaning that different word embeddings are created for each word in a sentence, capturing its content, also taking into account its position in a sentence. Furthermore, other researchers focused on Topic Modelling (TM) [36], [37] and semantic distances [38] to advance text representation learning. The approach of TM facilitates the identification of documents within the same category (i.e., same topic of discussion) that are partitioned into clusters (groups) [39].

Focusing on the classification models per se, deep learning-based methods are commonly used, including Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), as they allow the creation of effective text-based classification frameworks [40]. CNNs were originally created to improve image processing and have led to ground-breaking results in identifying items from a given list of objects [41]. In addition to CNNs, Recurrent Neural Networks (RNNs) [42] and especially Long Short-Term Memory (LSTM) networks, have been extensively used (either alone or in combination with CNNs) in the same direction [43], [44]. Compared to CNNs, RNNs are able to process temporal information, namely data that comes in sequences, such as sentences. Moving into the field of Natural Language Processing (NLP), they have been leveraged to improve performance across a wide range of tasks, including for instance text classification [45] and sentiment analysis [46].

A framework for CTI extraction and classification that leverages a CNN to identify the domain where the CTI could be assigned (i.e., Healthcare, Power Grid, General), while also utilising IoC extraction approaches to identify undetected types of IoCs has been proposed in [47]. Subsequently, the generated IoCs and their respective domain tag (identified with CNN) were used to create a categorised CTI with a specific domain. Finally, a classification approach that extracts CTI originating from hacker forums is presented in [48] by utilising two different variants of RNNs, namely Gated Recurrent Unit (GRU) and LSTM, resulting in high accuracy.

a: NAMED ENTITY RECOGNITION

Named Entity Recognition (NER) is the process of locating and classifying named entities mentioned in unstructured text into predefined categories (e.g. people, organisations, locations, expressions of times, quantities, monetary values, percentages, etc.), thereby facilitating the understanding and organisation of large volumes of text. NER plays a crucial role in various fields, including machine translation, question-answering systems, and information retrieval [49].

Different approaches can be used for NER, including *knowledge-based* methods, which rely on predefined rules to extract named entities, *feature engineering-based* methods, where entities are extracted automatically from a text without depending solely on rules, as well as *deep learning-based* methods. Depending on the data annotation procedure being followed, feature engineering-based methods can be categorised into unsupervised methods (which rely on data similarities), semi-supervised methods (employing strategies like co-training and self-training), and supervised methods (using techniques such as Hidden Markov Models). Deep learning-based methods often utilise Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or a combination of both [49].

RNNs and variations of LSTM network approaches, such as BiLSTM-CRF model which combines a bidirectional LSTM network with a Conditional Random Fields (CRF) layer, constituted until recently the most promising solutions concerning NLP-related tasks [50], [51].

In 2018, ground-breaking releases, such as Bidirectional Encoder Representations from Transformers (BERT) models [52], [53], brought radical and thorough solutions to language-based problems and as a result to NER tasks. BERT is pre-trained on WikipediaCorpus[11] of 2, 500 million words and BookCorpus[12] with 800 million words.

In our work, we use transformer-based models, such as BERT, as they allow the identification of many cybersecurity entities and classify them into the appropriate category with high accuracy [54]. In general, models like BERT consist of multiple encoder and decoder layers that utilise an attention mechanism. This architecture is generally known as transformer-based architecture. The attention mechanism assigns importance to inputs that have undergone a linear transformation, focusing on key elements to produce the output. A notable strength of these models is their ability to perform effectively without extensive labelled data. Initially, they are trained using an unsupervised approach, and subsequently, a smaller dataset is employed for supervised learning during the fine-tuning phase. Models following the

[10] https://machinelearningmastery.com/multi-label-classification-with-deep-learning/

[11] https://huggingface.co/datasets/wikipedia

[12] https://huggingface.co/datasets/wikipedia

above architecture can obtain a deep perception of language structure and architecture.

2) DATA CORRELATION

With regard to data correlation, several studies have attempted to tackle the issue of identifying and linking attacks within cybersecurity-related data. Attack identification is a very important procedure for individuals and organisations as can help them protect from cyber attacks. Data correlation is also a crucial procedure as it provides individuals and organisations with a broader view of the various aspects of an attack. For example, an IP address identified in log data performing malicious actions can also be found in external data sources executing other malicious activities. Thus, obtaining further knowledge about a specific IoC or actor aids in enhancing cyber protection. For example, Fusion Hidden Markov Models (FHMMs) have been used to model attackers' behaviour using log data captured by a Cowrie honeypot [55].

In [56], the authors attempted to identify correlations in log data using a system called GroupTracer. The system specifically focuses on extracting TTP (Tactics, Techniques, and Procedures) profiles and identifying potential attacker groups behind complex attacks targeting IoT systems. The process involves capturing attacks using IoT honeypots, extracting relevant information from logs, and automatically mapping the attack behaviours to the ATT&CK framework to extract TTP profiles. However, their mapping approach was not scalable due to the use of static rules.

According to the literature, it is evident, that significant advancements have been made in terms of utilising CTI towards the enhancement of organisational security. Honeypots, particularly low and medium-interaction types such as Cowrie and Dionaea, play a significant role in capturing attack tactics and techniques. Most approaches deploy honeypots implementing Docker containers, while the collected data is analysed on platforms that facilitate analysis, such as MISP. Concerning high-interaction honeypots, whilst providing comprehensive information, they require complex deployment and maintenance efforts. Furthermore, valuable insights into emerging cyber threats can be extracted from various web layers by leveraging web crawlers. Consequently, there are limited frameworks in the literature that leverage high-interaction rather than low or medium-interaction honeypots.

Aiming to both improve and enrich the quality of the CTI data, several research efforts have been made towards the enrichment through data classification and correlation. Techniques such as text classification, topic modeling, and deep learning models like CNNs and RNNs, along with advanced language models like ELMo and BERT, are commonly used for CTI extraction and classification. Named Entity Recognition (NER) aids in organising large volumes of text data, while data correlation techniques link attacks within cybersecurity data to provide a comprehensive view of threats. By leveraging these advanced techniques, organisations can enhance their security capabilities and effectively mitigate cyber threats.

Compared to the existing approaches in the literature, the ThreatWise AI framework integrates Wazuh[13] agents to collect internal data and uses a customized MISP[14] platform for managing and sharing Cyber Threat Intelligence (CTI). This creates a fully automated, streamlined process for CTI management. What sets ThreatWise AI apart is its novel outlier detection module, which ensures high data quality by distinguishing between normal behaviour and potential malicious activity.

Unlike traditional approaches that typically rely on a single platform, ThreatWise AI gathers data from both X (formerly Twitter) and Reddit. It employs an evasive crawler that mimics human behaviour to access dynamically loaded content. Additionally, the framework leverages advanced machine learning algorithms to enrich CTI by correlating data from internal and external sources.

By introducing these innovations throughout the entire CTI lifecycle– from data collection and analysis to storage, enrichment, and sharing– ThreatWise AI offers a more holistic solution than existing studies, which often focus on only one aspect.

III. THREATWISE AI FRAMEWORK

This section presents the proposed ThreatWise AI framework providing a detailed description of the overall framework as well as the different novel tools, concerning CTI gathering, extraction, enrichment and sharing. Furthermore, it presents a high-level description of the developed tools that are integrated into such a framework.

The proposed ThreatWise AI framework provides a holistic approach towards the gathering, analysis, enrichment, and sharing of CTI in an efficient and secure manner. Overall, it consists of several components which enable (i) the collection of information regarding cyber threats and vulnerabilities from several sources, (ii) CTI extraction from the gathered data, (iii) correlating the extracted CTI, (iv) storing, and (v) sharing the extracted CTI among other third parties such as CERTs and organisations in a secure manner. The high-level architecture of the ThreatWise AI framework and its components is presented in Fig. 1.

Initially, data are gathered from internal and external sources and are analysed to identify and extract Personal Identifiable Information (PIIs). Next, the data from external sources are classified by leveraging the Data Classification Model which classifies the data in terms of their relevance to either the cybersecurity or CTI domain. Subsequently, the resulting content is analysed to identify information such as IoCs and IoAs, to create the CTI entry. The resulting CTI is stored and correlated using both simple and advanced correlation methods, resulting in enriched CTI that is shared using a MISP instance that is configured according to the needs of

[13] https://wazuh.com/

[14] https://www.misp-project.org/

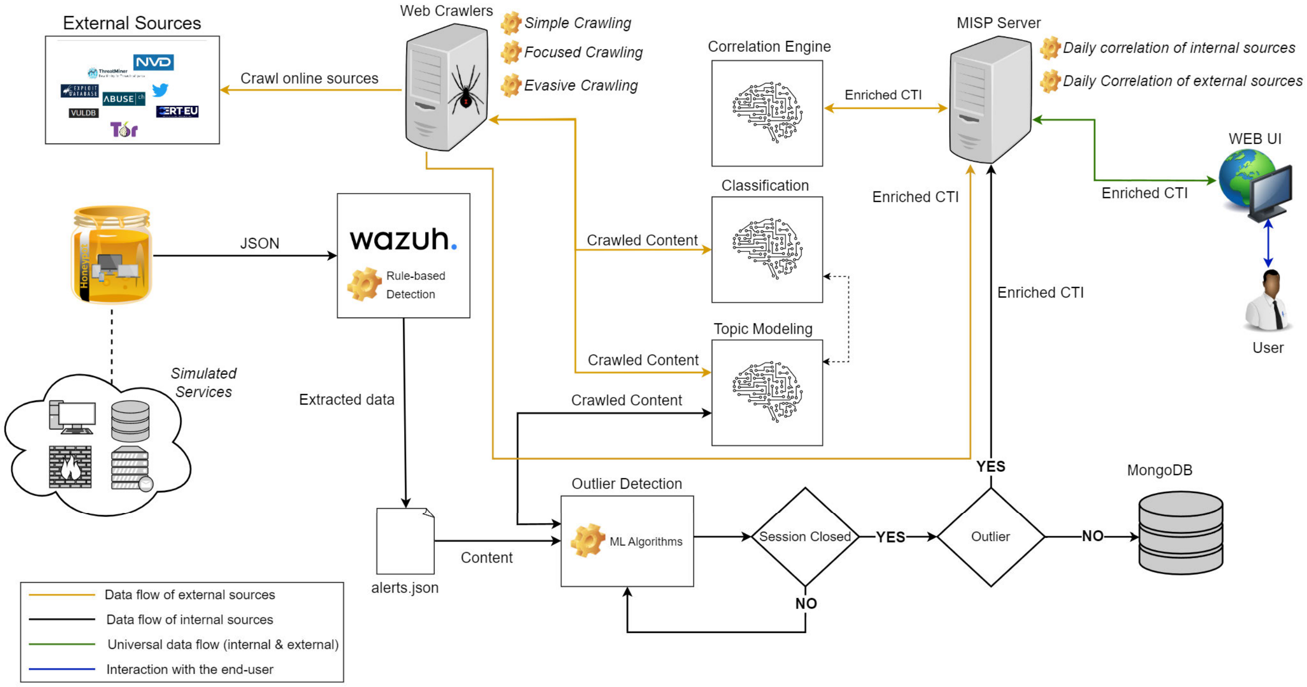


FIGURE 1. ThreatWise AI high-level architecture.

the framework. Simple correlations concern the correlation of threats based on similar values in different fields such as the same source IP from where the attack has originated. The simple correlation of the proposed framework is based on the default correlation engine of MISP.[15] The MISP correlation engine aims to identify relationships between attributes among the stored MISP events. In particular, MISP leverages a rule-based approach to examine if a certain value of an object's attribute (e.g., same domain name) exists in other events.

On the other hand, advanced correlations correlate threats based on different features that are extracted from the identified threats. Each functionality of ThreatWise AI is performed by separate tools that have been developed as part of the framework. Specifically, We put our focus on the application of text similarity methods to CERT feeds in order to identify documents that refer to the same threat or relevant threats. Our text similarity module takes as input a web document and gives a list of the most relevant documents to this initial output. The proposed ThreatWise AI framework has been tested and applied across various domains, including naval, smart city, and healthcare sectors. In each domain, it has demonstrated improvements in efficiency, resource management, and overall effectiveness. These promising results highlight the framework's versatility and potential for broad applicability, indicating its capability to deliver impactful outcomes in diverse domains and settings.

A. INFORMATION GATHERING

ThreatWise AI includes a novel tool responsible for gathering cybersecurity-related data from external and internal sources and proceeds with analysing the content to extract CTI. Subsequently, the extracted CTI is stored on the modified MISP instance.

With regard to the external sources, ThreatWise AI leverages three different crawling approaches which provide specific functionalities. General crawling simply extracts the text content without applying any filtering to the data. On the other hand, focused crawling leverages ML algorithms for data classification which enables the content filtering of the target web page to store only data that are relevant to cybersecurity or CTI. While focused crawling provides more advanced capabilities, it cannot address the issue of some web pages displaying different (limited) content to crawlers compared to web browsers. Therefore, the third approach concerns a novel evasive crawler which imitates human user behaviour allowing it to scrape content that is loaded dynamically or content that is displayed on the browser instead of the content that would be available to a crawler.

Regardless of the web crawling approach that is selected, all data is examined by a domain classification algorithm which classifies the documents either in one of the possible domains of interest (i.e., aviation, power grid, and naval) or labels it as general cybersecurity content. The domain classification functionality is performed after the initial classification function and enables further exploitation of the selected documents for scenarios and purposes that are domain-specific.

[15] https://www.misp-project.org/features/

The gathering from internal sources is achieved by monitoring a Dionaea honeypot cloud VM that is deployed on the Amazon Web Services (AWS)[16] platform. A Wazuh agent is deployed on the honeypot instance and monitors the various logs of the system in real-time. Once a new entry is added to these logs, the agent forwards it to the Wazuh server which is located in a remote location and stores the received logs in a file called *alerts.json*. The Wazuh manager parses the entry in a rule-based manner with the aim of evaluating that the entry concerns a cybersecurity incident and stores it in an *alerts.json* file. Subsequently, the newly added content is forwarded to the outlier detection module.

ThreatWise AI utilises a rule-based approach to examine the gathered content with the aim of extracting CTI information (i.e., IoCs and IoAs) which will be included later in the MISP event. Following the CTI extraction process, the content is enriched by applying appropriate MISP taxonomies according to the identified information. Furthermore, the outlier detection module leverages ML algorithms to identify whether new behaviours in honeypots belong to the same distribution as existing behaviours (i.e., inliers), or should be considered different (i.e., outliers). The analysis process includes the grouping of the identified incidents into sessions and for each session, several measurable values (i.e., features) are extracted. Examples of such values include the duration of the session (in minutes) divided by the number of connections, the number of total web pages requested, and the number of HTTP GET/HEAD/POST/CONNECT requests. The utilised extraction approach is analogous to the web bot detection problem [57], [58], considering that the honeypot instance also simulates web services.

The logs are initially split per IPv4 address and when an IP remains idle for more than thirty (30) minutes, the session is closed and the next entry from this IP results in the creation of a new session iliou2019towards, [58]. After closing the session, the outlier module examines whether the session is categorised as an outlier or an inlier. Concerning the latter case, the session is stored in a MongoDB instance whereas in the case of an outlier session, it is stored on the MISP instance.

B. INFORMATION SHARING

The extracted CTI data from all sources are stored on MISP in the form of MISP events. MISP supports a variety of different functionalities including detection, storing, and sharing of both technical and non-technical information concerning malware samples, incidents, attackers, and other relevant cybersecurity information. Furthermore, MISP supports data export in a variety of formats such as the Structured Threat Information eXpression (STIX)[17] which enables the interoperability of the platform with other cybersecurity tools (e.g., SIEM).

Each MISP event includes objects and attributes in accordance with the content of the stored CTI entry. The stored data is further enriched by utilising simple and advanced correlation on a daily basis. Simple correlation leverages the correlation engine of MISP to correlate threats based on similar values in different fields such as the same source IP from where the attack has originated, whereas advanced correlation utilises ML algorithms to correlate threats based on various features that are extracted from the data. Finally, the stored enriched CTI is available to the end-user via the Web User Interface (WEB UI) of MISP.

IV. DEVELOPMENT OF THE FRAMEWORK COMPONENTS

This section presents the development of the proposed ThreatWise AI framework and the relevant components and sub-components, concerning CTI gathering, extraction, enrichment and sharing. In particular, ThreatWise AI incorporates different sub-modules that have been developed as part of the framework, each of which is responsible for different functionality with respect to the CTI collection, enrichment and sharing, being also interoperable with each other.

First, the cybersecurity data are gathered and analysed from internal and external sources in order to identify and extract IoCs and IoAs to create the CTI entry. Subsequently, the extracted CTI is correlated using both simple and advanced correlation methods, resulting in enriched CTI that is shared using a modified MISP instance. Simple correlations concern the correlation of threats based on similar values (e.g., IPv4 addresses) in different events. On the other hand, advanced correlations correlate threats by leveraging ML algorithms. The developed ML models and the data that has been used for their training are also described in this section.

A. INFORMATION GATHERING

Information Gathering involves the gathering of cybersecurity information from both internal (i.e., honeypots) as well as external (i.e., online) sources. The information is gathered and stored in a central location for subsequent analysis. This section provides a comprehensive description of the different crawlers that have been developed that allow collecting data from the Surface, Deep, and Dark Web, as well as from social media platforms. Furthermore, the section describes the selected honeypots along with their functionality, challenges and deployment.

A search engine can access the contents of the Surface Web which consists of the web that the internet users use on a daily basis; it can be defined as the layer with which the users mainly interact on a daily basis. Deep Web is considered a special category of the Surface Web, where its content is not directly accessible from search engines and is available solely via other interfaces. Common examples of Deep Web domains are Facebook and Reddit, government resources, and academic content [59]. Finally, the Dark Web is defined as the lower layer of the internet. It includes

[16] https://aws.amazon.com

[17] https://oasis-open.github.io/cti-documentation/stix/intro.html

hidden content, intended mainly for illicit purposes, and is typically accessible only using special software like the Tor browser[18] [60].

1) CRAWLERS

Aiming to support the data collection from the different types of external sources, three major tools have been developed:

- **Web Crawler:** This crawler enables the collection of content from forums and web pages of interest from both the Surface and the Dark Web.
- **Custom Crawlers:** Custom crawlers are utilised to support the data collection from semi-structured or structured sources, such as vulnerability databases and CERT feeds (e.g., EU-CERT). The semi-structured nature of these sources requires a different approach than the forums and web pages. In particular, the content of semi-structured sources does not have a well-defined structure which can be parsed with existing parsers. Therefore, custom scripts, adapted to the structure of these sources are required in order to properly analyse the content.
- **Social Media Crawler:** This crawler leverages the respective official API of each social media platform for the collection of relevant social media posts.

The different types of crawlers not only enable the data collection from heterogeneous sources but also facilitate the extraction of CTI from the gathered content. The crawlers parse the content in order to identify IoCs and IoAs that are extracted to subsequently compose the CTI content.

The Web Crawler comprises three types of crawlers, each supporting a unique set of functionalities. In particular, it includes: i) a General Crawler which simply extracts the text content of a target web page and follows all the included hyperlinks; ii) a Focused Crawler that collects web pages that are relevant to the cyber-security or the CTI domain; and iii) an Evasive Crawler which imitates human user behaviour allowing to scrape content that is loaded dynamically or content that is served from websites when browsed by humans instead of crawling bots.

a: WEB CRAWLER

The Web Crawler is responsible for navigating through the web by following links found in web pages (provided by the user) and parsing their content to gather information [61]. The crawler is comprised of three sub-modules, namely the *Frontier*, the *Fetcher*, and the *Parser*. Each sub-module is responsible for specific tasks of the Web Crawler, facilitating the management and maintenance of the tool. The architecture of the crawler is illustrated in Fig. 2.

In particular, the *Frontier* contains the list of URLs that are identified and will be downloaded. Initially, it includes the seed URLs (i.e., the URLs that are provided as input to the crawler by the user) and updates its list with new URLs found through several iterations.

Then, the *Fetcher* is responsible for downloading and parsing the target URL. The Fetcher uses several iterations to remove URLs from the frontier and download their content. To enable the crawler to traverse both the Surface and the Dark Web seamlessly, the Fetcher utilises a web proxy service that is responsible for forwarding each web page to the respective Dark Web service.[19] The process is repeated until the desired depth (i.e., a maximum distance between the seed and crawled web pages) has been reached, or a maximum time duration has passed, or there are no more available web pages to download. Finally, concerning the *Parser*, it is responsible for extracting the hyperlinks that are contained in each web page and for subsequently forwarding them to the Frontier.

The content that is gathered by the Web Crawler must be relevant to either the cybersecurity or the CTI domain. Other irrelevant content is unlikely to include CTI information or in case it includes, it could contain limited information. Contrarily, cybersecurity content has increased CTI value given that most of the time it includes technical information which can be easily converted to CTI (e.g., malware names, hashes, attack group names). The filtering of the gathered data results in reduced noise (i.e., irrelevant content) thus facilitating the later processing by the ML algorithms. Furthermore, the quality of the stored data is improved since the data is relevant to the cybersecurity domain. In particular, the gathered data must include high-level or more technical information (e.g., vulnerabilities, attack methods, information about the attackers). Therefore, the developed IG component integrated a focused crawler which enables the filtering of the gathered data according to the target domains.

b: FOCUSED CRAWLER

The collection of CTI-related information introduces a significant challenge considering that while a web page contains information relevant to cybersecurity, it might not include any CTI-related or even cybersecurity information. Consequently, the web crawler should support a filtering functionality to distinguish the relevance of the web pages to the desired domains (i.e., CTI and/or cybersecurity). In this regard, the filtering functionality is provided by the *Focused Crawler* that has been developed and integrated with the Web crawler [62]. As depicted in Fig. 2, the general crawler includes three different sub-modules, the *Frontier*, the *Fetcher*, and the *Parser* while the focused crawler also includes a link selection process which enables the crawler to follow only hyperlinks that lead to relevant resources. In general, in a focused crawler system, the selection of the hyperlinks to follow is guided by automatic text classification methods that may be applied to the following types of input and combinations of them [63]:

- Local context of a hyperlink;
- Global context of the hyperlink's parent page; and
- Global context of the destination page of the hyperlink.

[18] https://www.torproject.org

[19] https://www.privoxy.org

Using the first two types of input is considered the most efficient way in terms of time since the destination web page does not have to be downloaded. For example, concerning the first method, only the hyperlink is used by the classifier to decide whether it leads to a web page with a relevant topic or not. Despite this advantage, the first two methods also introduce some disadvantages. Concerning the local context, it poses the risk of not containing sufficient evidence to decide if this web page is relevant [63]. In addition, while the second method is based on the assumption that parent web pages tend to link to others with similar content, this might not always be the case for many web pages. Concerning the efficiency of this method, it is found to be relatively more inefficient in terms of time, given the challenging particularities of our specific application, which hinder the decision-making concerning the relevance of a web page a difficult task. On the other hand, a classifier that bases its decisions on the text content of the destination web page is more effective in terms of text classification since it provides more relevant results. Consequently, we implemented this approach within the *Focused Crawler*.

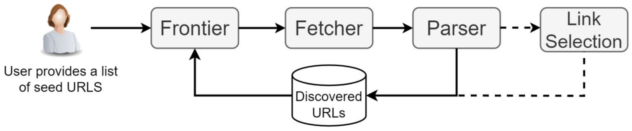


FIGURE 2. General and focused crawling architecture.

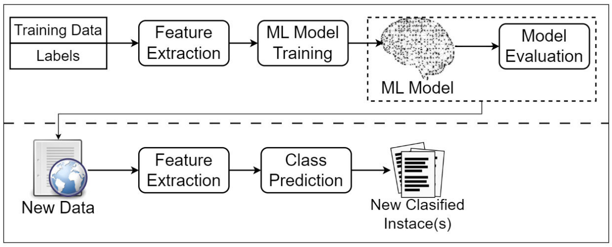


FIGURE 3. General pipeline of the supervised text classifier.

The focused crawler of IG uses a text classifier that is based on our previous work [64] which leverages supervised ML methods. The classifier developed in this work was able to distinguish three classes of web articles: i) the CTI-related, ii) the cybersecurity-related (i.e., articles that are relevant to the broader topic of cybersecurity but that do not contain strictly CTI-related content such as specific IoCs) and iii) those that are not related to cybersecurity.

On the contrary, we have found that the majority of web articles that are related to cybersecurity issues in specific industries such as aviation, naval or healthcare have either no CTI content at all or contain little technical information to be strictly characterised as CTI-related, although their content could be useful for further analysis. For this reason, we decided to use a classifier that is able to distinguish all three categories but filter out only the articles that fall into the no-cybersecurity-related class. The training set will be described in section V.

The general pipeline of the classifier is presented in Fig. 3. As depicted in the architecture, in the training phase an ML-based classification algorithm (i.e., Support Vector Machine) receives as input a set of labelled documents, each represented as a numerical feature vector. The application of the algorithm on the training set results in a model that accurately describes the relationship between the documents' features and their labels (or classes). The learnt model can then be applied to new documents, represented with the same feature set, to predict their respective classes.

Before converting documents to machine-readable format (i.e. feature representation), preprocessing should be performed to reduce noise while preserving the most important parts of them. We followed the same procedure as in [64], in which the first step is the removal of the irrelevant content (such as advertising blocks, templates, footers, etc), called boilerplate, from each web page so that only the main article remains to be used for the next step. For this, we have used the Readability tool.[20] Boilerplate removal is followed by a step of tokenisation, whereas then we applied stopwords removal and the so-called "CVE normalization", which converts all mentioned CVE IDs into a unique placeholder term. The combination of the last two steps resulted in the best performance among several possible steps and their combinations thereof (including, for example, stemming), consistent with our previous work when the SVM classifier was used, although the differences in the performance were quite small. For documents' representation as vectors, the Bag of Words (BoW) model and the frequently-used TF-IDF weighting scheme were used.

Finally, the learning algorithm that was applied to the vectorised documents is linear SVM [65], [66], which yielded the best results when compared to Random Forest as depicted in [64].

c: EVASIVE CRAWLER

While the general and the focused crawlers are able to crawl most web pages, some websites might introduce some peculiarities that hinder the crawling process. In particular, some web pages provide dynamic content while also displaying different content to human visitors that use legitimate browsers compared to the content that is available to web crawlers. Furthermore, some web servers provide different content to crawlers and browsers compared to the content that a human would see when visiting these web pages. Therefore, an evasive crawler has been developed which exhibits browser fingerprints and human-like browsing behaviours, thus crawling the target web page in a human-like manner. In essence, evasive crawling enables the gathering of the content that would be available to a human who is visiting the web page. The gathering of such information is particularly

[20] https://pypi.org/project/readability-lxml

useful in case of vulnerabilities (e.g., web pages from the dark web) that can facilitate the enrichment of the collected threat intelligence. It is important to note that the evasive crawler is not used to gather data from web servers that explicitly mention that they do not want their content to be crawled (i.e., through the robots.txt file).

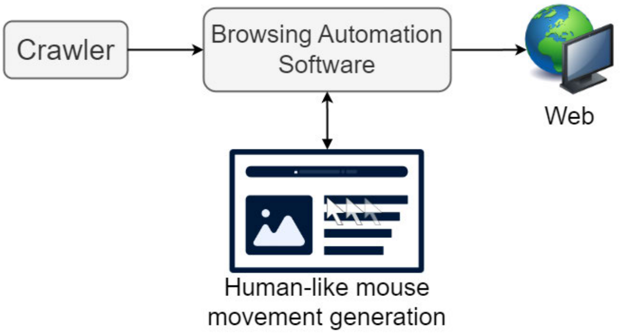

Crawler
Browsing Automation Software
Web
Human-like mouse movement generation

FIGURE 4. Evasive crawling architecture.

The architecture of the evasive crawler is presented in Fig. 4. The web crawler allows the enabling of the "evasive crawling" mode upon adding a new source to be crawled. The evasive crawling module enhances the functionality of the crawler by introducing two additional features: (i) a browser fingerprint, and (ii) a human-like browsing behaviour. Both features can be generated using browsing automation software such as Selenium,[21] which is utilised by the evasive crawler. Selenium has been selected since it enables the effortless creation of advanced web bots that support the majority of features of common browsers while it also allows bots to interact with the server in a human-like manner (i.e., perform mouse movements, fill in forms, click elements, etc.).

With regard to the fingerprinting functionality, Selenium in the evasive crawler has been enhanced to achieve evading detection and improve efficiency. Among the most common ways to detect Selenium is through its JavaScript variables which are not included in common browsers [67]. However, the variables that need to be updated depend on the respective driver that will be utilised. Selenium is configured to use the Opera drive whereas the JavaScript variables that have been updated are:

- JavaScript variables that include the prefix "cdc_" (the following characters of those values vary);
- JavaScript variables with value names that contain the words "selenium" or "webdriver".

According to relevant research [67], [68], these configurations are sufficient to evade detection based on a fingerprint.

Finally, the evasive crawler supports three different behaviour modes:

- **Repeat Mode:** In this mode the mouse trajectories of real humans performed on a specific web page are added to the crawler and used/repeated when the crawler accesses this web page.
- **Heuristics Mode:** The crawler performs human-like mouse movements utilising heuristic techniques. Examples of human-like mouse trajectories are depicted in Fig. 5.
- **Advanced Mode:** Advanced mode leverages Generative Adversarial Networks (GANs) to artificially generate human-like mouse movements.

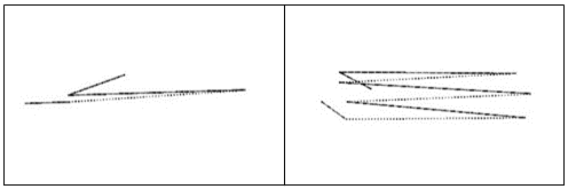


FIGURE 5. Examples of human-like mouse trajectories.

d: SOCIAL MEDIA

Security vendors, experts, and specialists often discuss on social media about various vulnerabilities, cyber-attacks and other cybersecurity incidents. X is considered a social media platform with the most active discussions regarding cybersecurity. X posts from accounts such as ExploitDB,[22] CERTEU,[23] and threatintelctr[24] provide useful information about threats and vulnerabilities.

Considering the value of social media content, the IG component, supports a social media crawler for information gathering from X and Reddit[25] platforms based on keywords provided by the user, while for X it also supports monitoring specific accounts.

With regard to X, the social media crawler leverages the official X API[26] by using tweepy library.[27] Tweepy is an open-source python library that provides a convenient wrapper to access the Twitter API via the Python programming language. Due to the rate limitation of the X API, when that rate limit is reached, the social media crawler switches to sleep mode. The collected data is stored on MISP as MISP events for further analysis to identify possible correlations and enrich the information.

Concerning Reddit, the crawler utilises the Praw[28] Reddit API wrapper which enables the retrieval of whole subreddits, preserving their original comment tree structure. The users can provide subreddit names (e.g., cybersecurity, malware) and the crawler proceeds with downloading the posts along with the relevant comments.

[21] https://www.selenium.dev

[22] https://twitter.com/ExploitDB

[23] https://twitter.com/CERTEU

[24] https://twitter.com/threatintelctr

[25] https://www.reddit.com/

[26] https://developer.twitter.com/en/docs/twitter-api

[27] https://docs.tweepy.org/en/stable

[28] https://praw.readthedocs.io/en/latest/

2) HONEYPOTS

Internal sources such as honeypots, are considered very useful for the collection of intelligence regarding threats and vulnerabilities that are currently not known to the public. Unknown vulnerabilities (Zero-days) can be detected through anomalies in the internal networks of organisations. Furthermore, several sources, such as web logs and system logs, can be utilised in order to map the entire behaviour of the attackers, including the TTPs that they follow to carry out the attack. Therefore, this enables the extraction of intelligence concerning the attackers' behaviour, which is not frequently changed as it requires more effort from them. Another important advantage of honeypots is that with proper deployment and isolation, there is no impact if they get compromised and malfunction, which minimises the risk of allowing further spreading of the attack in the legitimate services of the organisation's infrastructure.

An extensive review has been made regarding the available honeypot solutions in order to identify the ones that are suitable for this work. The selection of the appropriate honeypot solutions was made considering two important aspects: (i) the services that each honeypot solution provides and how these can facilitate our objectives, and (ii) the maturity of the honeypot solutions and their maintenance status (e.g., frequency of commits on GitHub). Upon reviewing several available honeypot solutions[29] in terms of their maturity and provided services, we identified three honeypots as the most suitable: Conpot,[30] Gaspot,[31] and Dionaea. Conpot and Gaspot provide services related to ICS. In particular, Conpot supports protocols such as *modbus*, and *S7comm* which are used in the ICS domain whereas Gaspot simulates an above-ground storage tank which is commonly used in the oil and gas industry and can facilitate the gathering of intelligence concerning more targeted attacks against the ICS domain. On the other hand, Dionaea was selected for the variety of services that are generic in nature and enable the gathering of more common attacks including brute-force and SQL injection.

The selected honeypots have been deployed and configured on three separate cloud Virtual Machines (VM) on AWS. Given that Gaspot is integrated with Conpot,[32] the latter has been deployed in the two out of the three VMs on AWS. The first VM operates according to the default template which emulates a *Siemens S7-200 CPU*,[33] whereas the other VM operates according to the "guardian_ast" template (Gaspot) which simulates a Veeder-Root Guardian Above-ground Storage Tank (AST) which is a monitoring system that is typically used to monitor fuel levels in tanks. Each honeypot has been configured in order to reduce the noise (data irrelevant to cyber-attacks) to the best degree possible as well as to store the logs in JSON format. The later configuration facilitates further processing for the identification and extraction of relevant data. Moreover, JSON format is machine-readable, thus enabling easier, faster and more accurate parsing of the logs' data.

B. CTI EXTRACTION TECHNIQUES

This section presents the data analysis techniques utilised to identify and extract CTI from the gathered data. The data is first classified as relevant or not to the cybersecurity or CTI-domain to reduce the "noise". Subsequently, a novel NER algorithm is used to identify named entities, facilitating the domain classification which classifies the information to one of the domains of interest (e.g., naval, aviation, healthcare) or none of them (other). Following the classification, the extracted data is analysed by the outlier detection module that groups the content into sessions. Upon the creation, the session is considered active, whereas later is terminated either by the attacker or due to being idle for too long. Each closed session is then analysed and categorised as either inlier or outlier.

1) DOMAIN CLASSIFICATION

Here, we introduce the classification framework that aims to organise data into categories of interest. In particular, the aim is to characterise data as related to the domains of *naval*, *aviation*, *power grid* or none of them (*other*).

Building on top of state-of-the-art methods, both in terms of the model itself and the representation of the textual data, a text-based classification framework has been developed that allows for an effective organisation of the input data into categories of interest. In the remainder of this section, we describe how the classification framework is constructed by providing details concerning the parts of the different components. Overall, our framework consists of three main components: (i) Text-based Network, (ii) Metadata Network, and (iii) Combined Network.

a: TEXT-BASED NETWORK

The *text-based network* considers as input the raw text. As mentioned, there are several choices of neural networks that could form the basis of our classifier. Given their proven good performance so far as well as after carrying out a set of experiments with different methods, we use Convolutional Neural Networks (CNN).

Text Preprocessing: Before any text is fed to the network, a set of preprocessing steps is performed to reduce noise. Specifically, URLs, digits and single-character words, as well as punctuation and special characters are removed.

Embedding Layer: We use ELMo to encode words in a machine-readable manner. In particular, we use pre-trained embeddings from a language model trained on a 1 billion word benchmark.[34]

[29] https://github.com/paralax/awesome-honeypots

[30] https://github.com/mushorg/conpot

[31] https://github.com/sjhilt/GasPot

[32] https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-gaspot-experiment

[33] https://cache.industry.siemens.com/dl/files/582/1109582/att_22063/v1/s7200_system_manual_enUS.pdf

[34] https://tfhub.dev/google/elmo/2

Convolutional Layer: The final layer is a CNN with 100 filters and kernel size 3, followed by one 1D global average pooling layer (usually applied after the convolutional layers to downsampling their input), and a dropout layer of $p=0.1$.

b: METADATA NETWORK

In the *metadata network* non-sequential data is considered. The way a user expresses themselves in writing is unique. However, focusing on an industry domain (e.g. aviation, oil) with a specific context of interaction, the way of sharing information and expression can often involve common characteristics among different users. To identify those characteristics that could be used for more effective categorisation of textual data in the domains of interest, in addition to the textual content considered through the text-based network, a set of hand-crafted stylometric features is also exploited to further improve the classification performance.

In particular, stylometric features of different granularity are extracted,[35] including (i) Character-based features (e.g., ratio of upper-cased characters, exclamations and number of digits to the total number of characters); (ii) Word-based features (e.g., mean number of characters per word and acronyms); (iii) Sentence-based features (e.g., mean number and standard deviation of words per sentence); (iv) Dictionary-based features (e.g., ratio of abbreviations and discourse markers to the total number of words in a text); and (v) Syntactic features (e.g., part-of-speech tags). The extensive list of extracted features is described in [69].

To employ the metadata neural network of the model we experimented with multiple architectures. We ended up using 1 fully connected (dense) layer of size 64. We use *tanh* as the activation function, as it performs well with standardised numerical data.

c: COMBINED NETWORK

Finally, we combine the *text* and *metadata* networks using a concatenation layer using a fully connected output layer (i.e., dense layer) with one neuron per class we want to predict and *sigmoid* as activation function.

2) NER

The contents of several sources of interest include text that contains CTI information in an unstructured format. The identification, extraction and labelling of entities (e.g., types of attacks, or names of hackers) that are relevant to cybersecurity-related categories facilitates the process of CTI extraction. The NER module that has been developed enables the extraction of a considerably large number of named entity types [51]. Initially, the NER module was tested with the following four distinct state-of-the-art models:

- BERT [70] adopts a multi-layer bidirectional transformer logic, instead of the legacy left-to-right, predicting randomly masked tokens and successive sentences. Transformer blocks reflect the high number of encoder layers. The lower layers encode local syntax (useful for part-of-speech tagging) whereas higher layers can extract complex semantics (aspects of word meaning that facilitate word sense disambiguation tasks) [35]. A classifier is a linear upper layer. BERT, released from Google AI Language, leverages WordPiece Tokenizer and defined vocabulary.
- RoBERTa, released from Meta[36] [71], is replication research on Google's BERT that executed multiple comparisons and presented some performance assets. RoBERTa highlights the importance of some key hyper-parameters as well as the size of the training data which is particularly crucial since it could greatly affect the final result.
- XLNet, which is released from Google/CMU [72], is considered as one the latest breakthrough. In particular, XLNet has been reported to outperform BERT in a range of NLP tasks. The model has been developed with the aim of enhancing some of the BERT's drawbacks and is considered to have been pre-trained according to a generalised auto-regressive way with respect to the bidirectional orientation that surpasses BERT's limitations.
- Electra [73], is an approach motivated as an attempt to mitigate a critical drawback of models like BERT, namely the high training computational cost.

After extensive experiments and based on the results achieved, the NER module supports two of the four state-of-the-art models described above, specifically the BERT and XLNet models.

a: DATASET AND PRE-PROCESSING

The development of the NER module required a consistent and domain-related annotated corpus. The most prevalent annotated datasets for NER in the CTI domain are the DNRTI [54] and the MalwareTextDB dataset [74]. The latter is a corpus of annotated malware texts (39 reports with a total of 6, 819 sentences) constructed in 2017 that considers only four entities. Due to its limited size and the small number of entities, the dataset poses the risk of leading to vague results when applied to unknown sentences from CTI reports and other sources. Consequently, we focused on the use of the DNRTI dataset. The DNRTI dataset was selected considering that it is the most comprehensive, detailed, and coherent cybersecurity-related dataset currently available, and thus, can lead to solid, strong, and concrete insights regarding CTI mentions in unknown text.

Data preparation and pre-processing are crucial steps in determining the optimal approach for the NER module. Manually annotated collections are difficult to create since they require domain expertise and an extensive amount of time for the annotation process. DNRTI is a large dataset released in 2020 that contains 175, 220 words, annotated in 13 different entity categories leveraging the IOB/BIO annotation scheme.

[35] Extracted using the AUPROTK library: https://github.com/joanSolCom/AUPROTK

[36] https://about.meta.com/

According to this scheme, each token in a sentence is labelled as: i) B-label (e.g., 'B-HackOrg') if the token constitutes the beginning of a named entity, ii) I-label (e.g., 'I-HackOrg') if it is located inside a named entity, but not positioned first, or iii) O-label ('O') in case it is not part of a named entity (i.e., it is outside of the entity). Pre-fixed training/validation and test sets are included in the released version of the dataset.

It was observed that the released version of the dataset contained some issues (i.e., bad lines and typos). To address them, all missing values were removed and the identified typos were corrected, whereas some defective entity names were also replaced with the correct ones. The final format of the training/validation sets consists of 157, 945 tokens overall (9, 180 unique), with 140, 526 tokens in the training set and 17, 602 tokens in the validation set. The 'O' class tokens are the majority (124, 739 tokens), whereas the 'B' class and the 'I' class correspond to 20, 143 and 11, 254 tokens, respectively. The pre-fixed training and validation parts of the dataset contain 4, 963 token appearances referring to 'Hacking Organisations', such as *Cobalt*, *LuckyMouse*, and *OceanLotus*. Specifically, 3, 845 tokens refer to 'B-HackOrg' whereas 1,118 to 'I-HackOrg' entities. Out of these 3, 845 'B-HackOrg' tokens, 477 are identified as unique. According to the context, 140 unique 'HackOrg' single tokens of the validation set include 283 different uses (labels) in the training set. This observation indicates that the annotation tags of the words are modified according to the context. Apart from these tokens, there are 4308 tokens that refer to malware names ('B-Tool' and 'I-Tool') including *PlugX* and *NetTraveler* among other categories.

Each model is accompanied by its own tokeniser as well as its own vocabulary, enabling the mapping of each token with a unique code. During the development stage, the limit regarding the length of sequences is set to 120 sub-words. The largest lengths of sequences of tokens in our data were detected in the interval 99 to 125 depending on the different tokenisers; sequence length of up to 512 is supported by BERT, whereas it is unlimited for XLNet. Concerning BERT, the maximum length of sentences was 115 tokens and the mean length was 46.68 tokens with a standard deviation of 27.65. For XLNet the values were 49.39 and 28.72, respectively. Longer sequences were shortened whereas shorter sequences were padded (post-tokens) to comply with the fixed defined size.

In most cases, tokenisers and models are offered in both cased and uncased variations, with uncased variants of the models widely considered to perform better. Nevertheless, our implemented strategy is oriented towards focusing on case-sensitive versions which were deemed more suitable for cybersecurity-related NER. Tokenisers also split complex words into pieces so as to be identified by the vocabulary. To address this issue, corresponding labels had to be multiplied accordingly during this splitting process.

Finally, Neural Network inputs do not refer to text but to numerical values. The tokenisers integrate core features that enable the conversion of input tokens to IDs (indices numbers), encoding representations according to their vocabulary. Furthermore, IBO/BIO labels are modified to integer numbers and subsequently special tokens are added. Attention masks are also created as an additional input array to the input IDs and labels. At the point where the dataset is divided into training and validation sets, the attention masks are aligned with their respective input IDs and labels. This alignment ensures that each element – the actual data and the padding – is correctly identified and used appropriately during both the training and validation phases Attention masks are composed of float numbers. These numbers signal to the model whether a given token is an actual one (1.0) or a padding element to be ignored (0.0). Subsequently, the data loaders are configured. During the training phase, data is shuffled using a random sampler, whereas, during the validation phase, data is loaded sequentially using a sequential sampler.

C. CTI ENRICHMENT AND SHARING MODULES

The extracted CTI is enriched by leveraging simple and advanced correlation techniques to identify possible relationships among the stored data. Simple correlation is achieved by utilising the correlation engine of MISP which searches for similar values in all MISP events. With regard to advanced correlation, ML algorithms are used to identify and extract more sophisticated relationships by analysing various values that are extracted from the stored data.

1) OUTLIER DETECTION MODULE

Logs collected from honeypots usually contain data that is not of much importance for CTI extraction, such as requests from simple scanning scripts that try to identify vulnerable servers on the internet. Therefore, in order to separate important logs from advanced attackers from logs that contain just simple requests with no CTI value, an outlier detection module was developed. The purpose of this module is to identify whether new logs from honeypots belong to the distribution of normal behavioural patterns (i.e., inliers) or if they should be considered part of new emerging attacker behaviours.

The collected logs are processed by grouping them into sessions. The logs are initially split per IPv4 address. When an IP address stays idle for more than 30 minutes (i.e., it does not perform any request during this time period), this session closes and a new session is initiated upon a new request from the same IP address. To avoid having too many open sessions (i.e., when an attacker does not perform further action after 30 minutes), a task runs every three minutes to identify such hanging sessions and close them forcefully.

When a session closes, several measurable values (features) are extracted. These features were selected by examining the literature for different problems that use the same protocols [57], [75] and by enumerating all protocols and request types that are supported in each honeypot. The extracted features are then used as input in the outlier detection module for the identification of outlier sessions. The outlier module uses the Isolation Forest outlier detection Algorithm [76], since it has shown the most promising results based on our experiments as described in Section V. The full list of features is presented in Table 1.

TABLE 1. Features used for outlier detection. Multiple features can be presented in one line, differentiated by the '/' symbol.

ID	Feature
1	Duration of session (in minutes) divided by the number of connections
2	The number of total Web pages requested
3-6	Number of HTTP GET/HEAD/POST/CONNECT requests
7-10	Number of HTTP requests with HTTP response code 2xx/3xx/4xx/5xx
11-12	Number of HTTP requests with version 1.1/1.0
13-20	Number of web requests that requested for a file of type HTML/PHP/JS/CSS/image/XML/TXT or JSON file/CFG
21	Number of web requests for the ".env" file
22	HTTP request that includes in the path the string "/shell?"
23	HTTP request that has no file extension
24	Requests that generated an HTTP error log
25-26	Successful authentication requests/Insecure connections to SSH
27-30	New connection/Timeout/Connection lost/Other request for BACnet
31-34	New connection/Timeout/Connection lost/Other request for S7comm
31-39	New connection/Timeout/Connection lost/AST commend/other request to AST
40-43	New connection/Timeout/Connection lost/Other request the Modbus
44-45	Accept/Free request to SMB
46-47	Accept/Free request to Mongo database
48-50	Accept/Free requests to SIP / SIP message
51-53	Accept/Free/Login requests to MSSQL
54-60	Select/Create/Drop/Dump/Declare/Execute/Use commands to MSSQL
61-62	Accept/Free connections to the UPNP
63-66	Accept/Free/Source/Subscriber requests to the MQTT
67-69	Accept/Free/Transfer requests to EPmapper
70-71	Accept/Free requests to Memcache
72-75	Accept/Free/Login/Exec requests to FTP
76-79	Accept/Free/Login/Exec requests to MySQL
70-81	Accept/Free requests to HP LASERJET printer
82-84	Accept/Free/Connect requests to the PPTP
85-86	Accept/Free requests to the TFTP

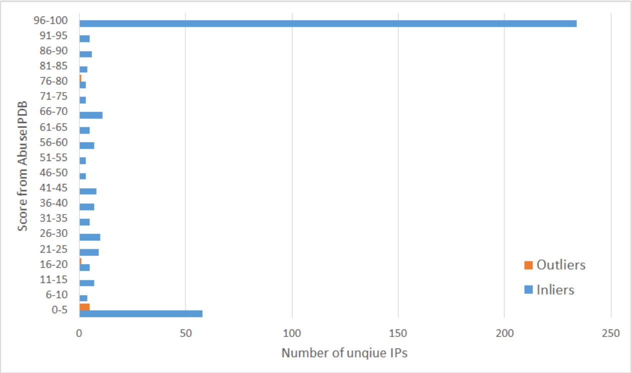


FIGURE 6. Scores from AbuseIPDB for inliers and outliers.

As it can be seen in Fig. 6, most sessions that were identified as outlier sessions by the module had a low score on AbuseIPDB. This means that these IP addresses have not been observed to be engaged in malicious acts, or that they are at the moment not classified as malicious. On the contrary, the IP addresses that have been classified as inliers have a very high score on AbuseIPDB. This means that they have already been identified as malicious at a great scale, probably through massive web scanning. This way, the outlier detection module is able to identify new attackers or malicious actors by focusing on outlier sessions that are not yet well known or have succeeded in staying undetected.

In the container, Watchdog[37] observes for changes in the *alerts.json* file of Wazuh. When alerts are detected, they pass through a session manager that groups them into sessions, as previously described. Once a session is defined, the features mentioned in Table 1 are extracted. These features are then passed in the trained Isolation Forest model, which classifies them as outliers or inliers. If a session is categorised as an outlier, all the logs that it contains are sent to the MISP[38] platform to be used as CTI. The sessions categorised as inliers are stored in a local database. All the logs and IPs of inlier sessions that are stored in the MongoDB (local) database are encrypted with the AES 256 CBC HMAC SHA512 algorithm. Celery[39] framework is used to schedule the necessary tasks within the outlier module, such as the task for force-closing the hanging sessions, as already described. Finally, another task is responsible for re-training the Isolation Forest model with the new logs that have been collected, once per day. This way, the outlier detection module manages to stay up-to-date and detect changes in behavioural patterns and the latest attack trends.

2) CTI CORRELATION TECHNIQUES

As mentioned, our approach uses two types of correlation, namely simple and advanced. Nevertheless, apart from this categorisation, the advanced correlation is further divided into the correlation of internal and external sources. Advanced correlation of internal sources aims to identify correlation among the information that is collected by the deployed Dionaea honeypot instance (e.g., SQL commands). On the other hand, advanced correlation of external sources enables the integration of information concerning cybersecurity incidents or vulnerabilities, reported from different external sources.

a: CORRELATION OF INTERNAL SOURCES

An advanced correlation module was introduced to enhance the quality of the extracted CTI and increase the functionality of the MISP correlation module. For the data collected from the honeypots (i.e., internal sources), an unsupervised Machine Learning (ML) technique called Association Rule Learning (ARL) [77] was used to extract higher levels of actions executed by attackers by identifying correlations

[37] https://pypi.org/project/watchdog/
[38] https://www.misp-project.org/
[39] https://docs.celeryq.dev/

among different commands. Correlations can identify relationships between attributes and indicators from malware, attack campaigns, or analysis.

Assuming that each IP depicts the action of one attacker, the advanced correlation module using the ARL technique could extract CTI from malicious attacks by investigating their behavioural/attack patterns. ARL is applied directly to the input data without extended pre-processing steps (e.g., specific feature extraction) thus facilitating the correlation procedure.

In general, ARL was initially introduced in commercial environments to model customers' purchasing behaviour [77]. The products that a customer buys can be considered distinct items and the objective is to identify items that are bought together. This procedure can be expressed in the form of rules. For example, a customer who buys a product also buys another product. The same logical procedure can be transferred to the cybersecurity domain. The command that an attacker executes can be considered as an item and the goal is the identification of items (e.g., commands) that an attacker executes together. In this way, attack patterns followed by attackers can be identified by correlating the executed commands.

The advanced correlation module follows the same logical procedure to find malicious actors' sequences of attacks or actions, using data collected specifically from host systems. Contrary to network data, host-based data contains richer attack information so better insights can be extracted from specific commands, run by attackers; this was first presented in a work of ours [78].

In ARL, the various commands that attackers run can be considered as items where the notion of an itemset describes a group of two or more items that appear together in a dataset. The If-Then structure is generally used in ARL to express correlations between different items. The if and then parts are generally known as antecedent and consequent, respectively. For example, if we take into account three different commands A, B, and C then a rule can be expressed in the form: If A and B, then C. The rule shows that commands C will be executed after commands A and B have been also executed [79].

An example of such an If-Then structure is provided below:
If: (frozenset('Drop Procedure DllUnregisterServer ',
c'Drop Procedure sp_password ',c'EXEC sp_OA',
'exec DllUnregisterServer '),

Then: frozenset("Create 'WbemScripting.SWbemLocator', objLocator OUTPUT ",'exec DllRegisterServer ')

In this rule, an attacker initiates actions by executing commands specified in the "If" section, followed by additional commands outlined in the "Then" section.

Initially, the attacker performs certain operations as detailed in the "If" part, including executing commands such as "Drop Procedure DllUnregisterServer," "Drop Procedure sp_password," "EXEC sp_OA," and "exec DllUnregisterServer." These commands indicate SQL database activities aimed at removing procedures and executing specific functions, potentially indicative of unauthorised or malicious attempts to unregister DLLs or alter stored database procedures. Following these operations, in the "Then" part, the attacker executes commands like "Create 'WbemScripting.SWbemLocator', objLocator OUTPUT" and "exec DllRegisterServer." These commands are associated with actions involving Windows Management Instrumentation (WMI), a Microsoft framework for managing data and operations on Windows operating systems.

In general, the commands that appeared in the "If" and "Then" parts of the rule, execute certain database and system commands based on the presence of other commands or procedures. If certain procedures related to system DLL registration and database password management are detected (via dropping them), then the script proceeds to create a WMI locator object and register a DLL. This could be part of a setup, maintenance, or security script in a database or system administration context. In ARL, different rules can be generated depending on the size of the dataset and the complexity of the commands executed by attackers.

b: CORRELATION OF EXTERNAL SOURCES

The correlation of textual data (i.e., web articles) is important in order to combine and integrate information provided from distinct sources. Through correlations, relationships between attributes and indicators from security incidents such as malware, and attack campaigns are identified and extracted. This intelligence can be very useful in the case of mapping different TTPs used by the same threat actors and enriching the CTI. For example, it allows gaining a more holistic view regarding a specific cyber incident reported in different sources or finding relations between different incidents or vulnerabilities.

Here, we describe the correlation of textual data, the purpose of which is to link two or more web articles based on their relevance. To assess the relevance between the collected web articles we use text similarity techniques. Specifically, we follow the probably most used approach for this task, which is to represent each document of our collection as a TF-IDF vector (similar to the text representation used in the text classifier of the focused crawler component) and then compute the similarity of any given pair of documents by measuring the distance of their vector representations [80]. We measure the distance between two vectors with their cosine similarity. Formally, if x and y are two documents and $\vec{v_x}$, $\vec{v_y}$ their respective vector representations, the cosine similarity of the vectors is defined as:

$$cossim(\vec{v_x}, \vec{v_y}) = \frac{\vec{v_x}^{\,T}\vec{v_y}}{\|\vec{v_x}\|\|\vec{v_y}\|} \tag{1}$$

The higher the cosine similarity score, the more relevant the two documents are. Given a document x from a set of collected documents, we retrieve the most relevant documents to x from this set by computing the cosine similarity scores of its vector with all the other respective vectors and ranking the documents based on their scores. In the context of CTI, the described approach has also been used in [81] and [82]. Importantly, we measure the effectiveness of using the TF-IDF representation and the cosine similarity in the task of retrieving relevant documents in the context of CTI, by collecting a real-world dataset of web articles. We present the dataset and the performance of our approach in Section III.

3) MISP

The extracted CTI from both the internal and the external sources is stored on the MISP platform which is one of the most widespread CTI sharing platforms. Moreover, the simple correlation of the proposed approach is based on the default correlation engine of MISP.[40]

MISP supports a variety of different functionalities including detection, storing, correlation, analysis and sharing of both technical and non-technical information concerning incidents, attackers, and other relevant cybersecurity information. Additionally, MISP supports a correlation engine that: (a) is able to identify relationships between attributes/objects and indicators from malware correlation engines, and (b) is capable of performing advanced correlations, such as fuzzy hashing (e.g., ssdeep[41]) or CIDR block matching. MISP stores data in a structured format known as MISP events, provides extensive support for cyber-security (including fraud) indicators for different vertical sectors (e.g., financial sectors), and provides a stable and secure environment for CTI sharing both manually and in an automatic manner using the MISP API.

To contextualise the collected information, the appropriate MISP data objects are used according to the content of the data that is stored. MISP provides the ability to overwrite, update, or replace objects according to the user's needs. Furthermore, users can create new objects in addition to the default ones, by defining the appropriate JSON schema of the new objects.

The extracted CTI from both internal and external sources is stored on the MISP server instance as a MISP event which includes the appropriate MISP data objects (upgraded when needed) according to the content of the stored CTI. In Table 2 we present the MISP objects utilised for our needs. Considering that the available MISP objects could not support all the required information that needs to be stored, a custom object with the name vulnerability-extended has been created in order to cover our data needs. The developed custom object is presented in Table 3.

V. EXPERIMENTS

This section presents the experiments that have been conducted on critical components of the ThreatWise AI framework's overall pipeline. The experiments enable the evaluation of the proper operation of each component as well as the overall quality of the relevant functionality.

TABLE 2. Utilised MISP objects.

MISP Object	Description	Attributes
Weakness	Weakness object describing a common weakness enumeration which can describe usable, incomplete, draft or deprecated weakness for software, equipment of hardware.	• ID • Name • Description
Exploit-poc	Exploit-poc object describing a proof of concept or exploit of a vulnerability. This object has often a relationship with a vulnerability object.	• Authors • Description • PoC • References
Twitter post	Twitter post (tweet)	Total number of tweets discussing about the specific CVE.
File	File object describing a file with meta-information.	• Entropy • Filename • Malware-sample • MD5 • SHA1 • SHA256 • SHA512 • Size-in-bytes
Command	Command functionalities related to specific commands executed by a program, whether it is malicious or not. Command lines are attached to this object for the related commands.	• Description • Location • Trigger
Credential	Credential describes one or more credential(s) including password(s), API key(s) or decryption key(s).	• Format • Origin • Password • Type • Username
Http-request	A single HTTP request header.	• Method • URL • User-agent
Network-connection	A local or remote network connection.	• Src-port • Layer7-protocol
Virustotal-report	Data from a VirusTotal report.	• Comment • Detection-ratio • Permalink

A. INFORMATION GATHERING

This section describes in detail the conducted experiment regarding the tools of ThreatWise AI that enable information gathering. In particular, the experiment aims to assess the cybersecurity classifier of the focused crawler.

[40] https://github.com/MISP/MISP/blob/2.4/docs/correlations.rework.md
[41] https://ssdeep-project.github.io/ssdeep/index.html

TABLE 3. Utilised MISP objects.

MISP Object	Description	Attributes
Vulnerability-extended	Vulnerability object describing a common vulnerability enumeration which can describe published, unpublished, under review or embargo vulnerability for software, equipment or hardware.	• ID • Description • Created (date-time) • CVSS-score • CVSS-string • CVSS-vector • CVSS-score2 • CVSS-string2 • CVSS-vector2 • Modified • Published • External references (multiple) • CPEs (multiple)

1) CYBERSECURITY CLASSIFIER OF FOCUSED CRAWLER

a: DATASET USED

The training set on which the cybersecurity text classifier was built is an enhanced version of the dataset used in [64]. The initial version consisted of 920 web pages from six cybersecurity-related, two technology-related and one generic news website. However, none of the domains of interest, namely the aviation, naval and power grid sectors, were covered as topics in this collection of labelled documents.

Consequently, an early attempt to classify text that covers these topics using a classifier built on the first version of the dataset suffered due to the discrepancy between the training data and the articles that the classifier has to predict. For this reason, we added 117 new documents that are related to one of the three domains of interest. Specifically, 59 articles related to the power grid, 32 related to the naval sector and 26 related to aviation have been annotated and added.

We tested the classifier on a test set consisting of **39** annotated web articles, all having as subject one of the three domains of interest. Specifically, **11** of the articles are related to the power grid domain, **16** to the naval industry, and **12** to the aviation sector. 12 of the articles are not cybersecurity-related, 21 fall into the cybersecurity class, and 6 are CTI-related.

b: IMPLEMENTED METHOD

As already discussed, we propose using a multiclass (3 classes) classifier and then merging the classes cybersecurity and CTI at prediction time, to finally get an output of whether the article of interest is cybersecurity-related or not. This was motivated by our finding that most of the mistakes of the classifier were between the CTI and the cybersecurity class. The confusion matrix presented in Table 4 depicts this.

c: EXPERIMENTAL SETUP AND RESULTS

In this section, we present the results of the binary classification scheme, for which the SVM algorithm was used, as already mentioned. We have experimented using 10-fold cross-validation with a few values for SVM's hyperparameter C and present the results of the model with $C = 4$. To evaluate we use the metrics balanced accuracy, precision, recall and the weighted f1-score as they are defined in section VI.

The first and last of them were selected as being appropriate to handle class imbalance in the test set since after merging the two cybersecurity-related classes the number of no-cybersecurity-related examples is 12 and the examples falling in the merged cybersecurity classes are 27. The experiments have shown a good overall performance for the classifier, which also achieves high levels of both precision and recall. The results are presented in Table 5. For reference, we also present the results of the multiclass classification scheme, where the classifier's predictions are used without any modifications. Finally, Figure 7 depicts the learning curve of the classifier (in the multiclass setting).

TABLE 4. Confusion matrix of the cybersecurity articles classifier for the case of three classes.

		Predicted		
		no-csec	**csec**	**CTI**
Actual	**no-csec**	9	2	1
	csec	3	16	2
	CTI	0	5	1

TABLE 5. Performance of the cybersecurity classifier, when used in the binary and the multiclass classification schemes.

	Balanced accuracy	Precision	Recall	F1-weighted
Binary	0.82	0.89	0.89	0.85
Multiclass	0.56	0.67	0.67	0.65

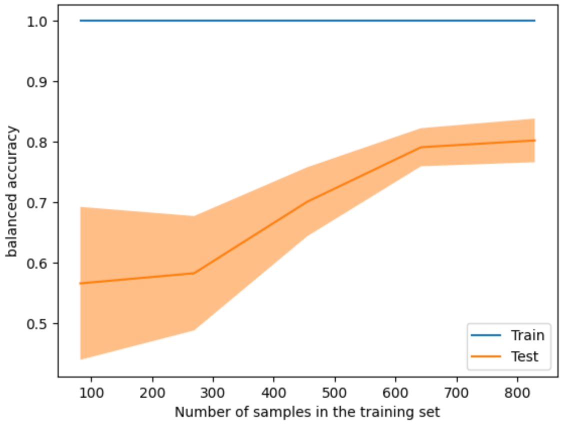


FIGURE 7. Learning curve of the cybersecurity classifier of focused crawler.

B. CTI ENRICHMENT

This section describes in detail the conducted experiment regarding data enrichment tools of ThreatWise AI. In particular, the experiment aims to assess outlier detection and pattern recognition.

1) OUTLIER DETECTION AND PATTERN RECOGNITION

a: DATASET USED

Closed sessions (i.e., idle for more than 30 minutes) were observed in honeypot instances. Specifically, we collected data from the Dionaea, Conpot, and Gaspot honeypots for the period of 15 days. The utilised dataset is presented in Table 7 where sessions refer to honeypot logs that are grouped per IPv4 address.

b: IMPLEMENTED METHOD

Concerning the evaluation of the outlier module, we used the Forest outlier detection algorithm by leveraging the same dataset gathered from internal sources. Two parameters are required by the Isolation Forest algorithm. The first one is the contamination percentage and the second is the number of base estimators to be used. For the selection of these two hyperparameters, the dataset was split into two parts. The first part includes the first 1011 sessions, whereas the second, which contains the other half, consists of 1011 sessions and was used as a test set.

c: EXPERIMENTAL SETUP AND RESULTS

Based on the quantitative evaluation of the results on the test set, the contamination percentage was selected to be 10% and the number of estimators 100. Furthermore, the PCA was used for dimensionality reduction and the number of features was reduced to two [83]. After the training of the Isolation Forest model with the training set, the test set was used to identify outlier sessions (i.e., sessions in which behaviour diverges from normal behaviour).

Results: Table 6 presents the number of sessions that were found to be outliers and the sessions that were found to be inliers. As shown in the dataset section in Table 7, the attackers that targeted the Conpot and Gaspot targeted more than one honeypot, indicating that these attackers are most probably automated scripts. Indeed, these sessions were found to present a similar behaviour with most sessions by the outlier detection module.

To get more insight into the sessions that were found to be outliers, the AbuseIPDB[42] service was used. It is a service that reports IP addresses, by giving a score, based on whether they have been observed to engage in malicious actions.

TABLE 6. Number of sessions and unique IPs for the inliers and outliers in the test set .

		Unique IPs				
	Sessions	Conpot	Gaspot	Dionaea	Total	Combined
Inliers	994	2	6	390	398	397
Outliers	17	0	0	7	7	7
Total	1011	2	6	397	405	404

C. DATA CLASSIFICATION

This section describes in detail the conducted experiments regarding data classification tools of ThreatWise AI. In particular, the experiment aims to assess the text-based, metadata, and combined network of the proposed text-based classification framework to conclude the most optimal one.

TABLE 7. Honeypot dataset.

Honeypot	Unique IPs	Sessions
Dionaea	1,235	2010
Conpot	15	4
Gaspot	45	8
Total	**1239**	**2022**

1) CLASSIFICATION FRAMEWORK

a: DATASET USED

To train and evaluate the developed framework, we need a ground truth dataset that indicates which textual data belong to a specific domain (i.e. naval, aviation, power grid, or other). The ground truth dataset used for this purpose consists of 974 articles in total, where 63 belong to the *naval* domain, 83 to the *aviation*, 60 to the *power grid*, and the remaining 768 to the *other* category.

b: IMPLEMENTED METHOD

To conduct our experiments we use Keras[43] with TensorFlow.[44] We run the experiments on a server that is equipped with one GeForce RTX 2080 TI GPU of 11 GB GDDR6 memory.

c: EXPERIMENTAL SETUP AND RESULTS

Overall, the ground truth dataset is split into training (90%) and test (10%) sets, maintaining the proportion of classes. From the training set, 10% is kept as a validation set. For training, we use categorical cross-entropy as the loss function and Adam (with *learning_rate*=0.0001) as the optimisation function. A maximum of 150 epochs is allowed, and the validation set is used to perform early stopping. Training is interrupted if the validation loss does not drop in three consecutive epochs, and the weights of the best epoch are restored. Table 8 overviews the performance of the developed classification model in the three experimentation phases. Overall, we consider three phases:

- *Text-based network:* Evaluation of the overall performance when only the text-based network is considered;
- *Metadata network:* Evaluation of the overall performance when only the metadata network is considered; and
- *Combined network:* Evaluation of the overall performance when the combined network is considered.

Results: Table 8 overviews the results concerning the performance of the developed classification model on the three aforementioned experimentation phases.

[42] https://www.abuseipdb.com/

[43] https://keras.io/

[44] https://www.tensorflow.org/

TABLE 8. Experimental results of the Domain Classification model.

	Prec	Rec	F1	Acc
Text-based network	0.970	0.969	0.967	0.969
Metadata network	0.811	0.816	0.804	0.816
Combined network	**0.989**	**0.989**	**0.989**	**0.989**

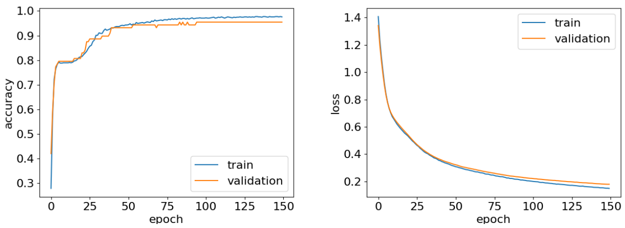
TABLE 9. Confusion matrix of the Domain Classification model (combined network).

		Predicted			
		other	naval	aviation	power-grid
Actual	other	77	0	0	0
	naval	0	7	0	0
	aviation	0	0	8	0
	power grid	1	0	0	5

When comparing text-based with metadata networks, from Table 8 we observe that the former performs better in all evaluation metrics. Although the hand-crafted stylometric features lead to adequate performance with about 81% precision and recall values, the text-based network with the ELMo method as the basis for word representation leads to significantly increased performance. As mentioned, ELMo, which is a contextualised word representation method, allows the modelling of complex characteristics of written word usage and expression by taking into account both syntactic and semantic information.

Overall, the best performance (98.9% precision and recall values) is achieved with the network that combines raw text and metadata (Combined network). Combining the text content with a wide range of stylometric features (metadata) allows for the characterisation of textual content at different levels, thus resulting in a better understanding of different semantics and the identification of underlying patterns.

Table 9 presents the confusion matrix of the combined network. As demonstrated, the classification model successfully assigns the majority of documents to their correct domains. Additionally, Figure 8 illustrates the accuracy and loss learning curves of the same model. With regard to the loss function, we observe a good fit, as both the training and validation losses decrease and stabilise at low values. Similarly, the training and validation accuracy curves indicate that the model generalises well, showing no signs of overfitting or underfitting.



(a) Accuracy (b) Loss

FIGURE 8. Learning curve of Domain Classifier: (a) Accuracy and (b) Loss.

D. NER

This section provides a detailed description of the experiments conducted to evaluate the NER module of ThreatWise AI. Specifically, the experiments with the BERT and XLNet models are described, along with the corresponding results.

a: EXPERIMENTAL SETUP AND RESULTS

As mentioned above the NER module supports two state-of-the-art models specifically the BERT and XLNet models. The DNRTI dataset used for the experiments entails pre-fixed sets for training and validation, as well as a holdout test set. The initial train/validation split is 89%-11%, while the train/test ratio is also 89%-11%..

During training different hyperparameters were used. The AdamW optimiser was chosen for training, along with weight decay as a regularization technique to mitigate overfitting by penalising large weight matrices. The fine-tuning process used 4 epochs and a batch size of 16. Other crucial parameters included a learning rate of 1e-4, a maximum sentence length of 120 tokens, and an epsilon value of 1e-12 for numerical stability. The maximum gradient norm was set to 1.0 to prevent exploding gradients. The models were trained with case-sensitive input (Lower Case set to False), allowing them to potentially capture nuances in capitalization.

Results: The results of the training are presented in Table 10. The two models achieved nearly the same results.

TABLE 10. Performance comparison of BERT base and XLNet base models.

Models	F1-Score	Precision	Recall
BERT base	**0.899**	**0.874**	0.928
XLNet base	0.898	0.870	**0.929**

The learning curves of the two models are presented below. The curves indicate signs of moderate overfitting after the third epoch.

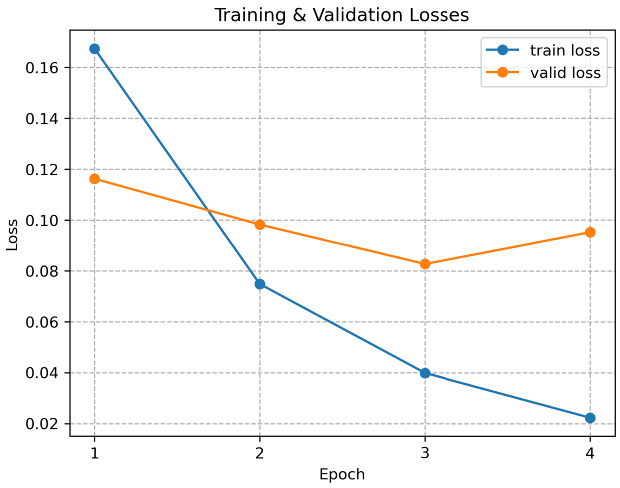


FIGURE 9. Learning curve of XLNet.

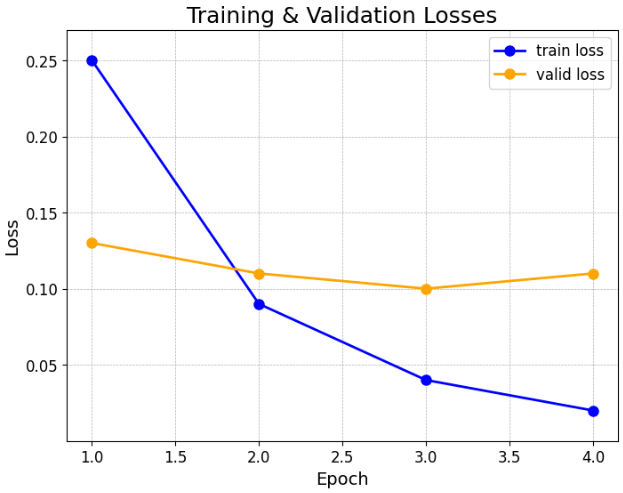
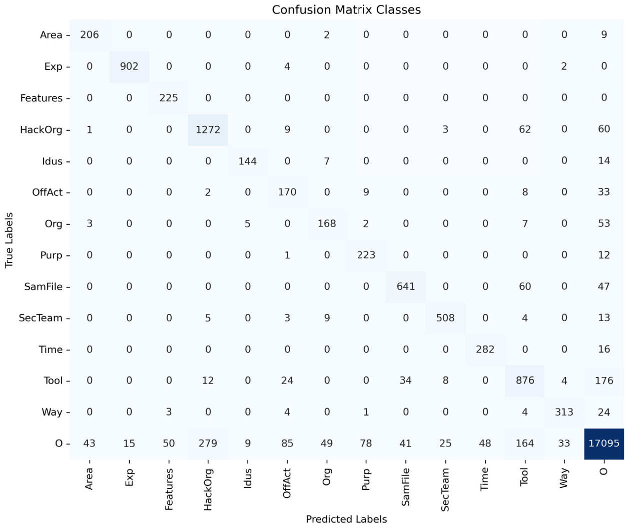


FIGURE 10. Learning curve of BERT.

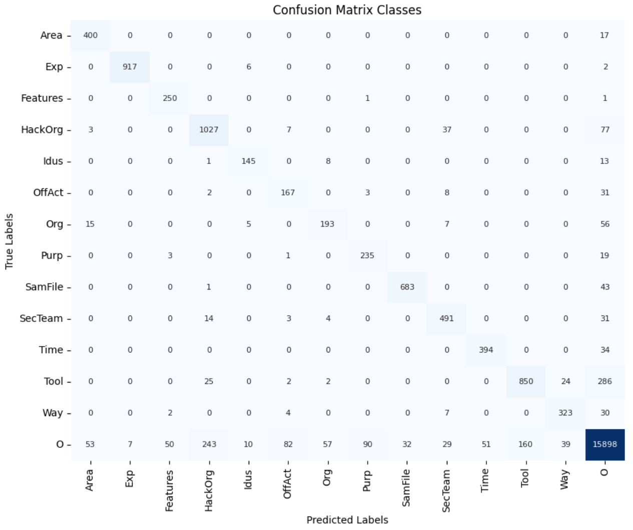
Last the confusion matrix of the models is presented below. The confusion matrix of XLNet depicts that the model has a strong performance in correctly classifying most of the categories, with particularly high accuracy in classes such as "HackOrg" (1272 correct predictions) and "Tool" (876 correct predictions). The matrix shows that XLNET generally maintains a high level of precision across various categories. The confusion matrix of BERT shows a similar pattern of strong performance, with high accuracy in classes like "HackOrg" (1027 correct predictions) and "Tool" (850 correct predictions). BERT also demonstrates a high precision across most categories.

In comparison, both XLNet and BERT exhibit strong performance in classifying the majority of categories accurately, with XLNet showing slightly higher correct predictions in some categories like "HackOrg" and "Tool".



	Area	Exp	Features	HackOrg	Idus	OffAct	Org	Purp	SamFile	SecTeam	Time	Tool	Way	O
Area	206	0	0	0	0	0	2	0	0	0	0	0	0	9
Exp	0	902	0	0	0	4	0	0	0	0	0	0	2	0
Features	0	0	225	0	0	0	0	0	0	0	0	0	0	0
HackOrg	1	0	0	1272	0	9	0	0	0	3	0	62	0	60
Idus	0	0	0	0	144	0	7	0	0	0	0	0	0	14
OffAct	0	0	0	2	0	170	0	9	0	0	0	8	0	33
Org	3	0	0	0	5	0	168	2	0	0	0	7	0	53
Purp	0	0	0	0	0	1	0	223	0	0	0	0	0	12
SamFile	0	0	0	0	0	0	0	0	641	0	0	60	0	47
SecTeam	0	0	0	5	0	3	9	0	0	508	0	4	0	13
Time	0	0	0	0	0	0	0	0	0	0	282	0	0	16
Tool	0	0	0	12	0	24	0	0	34	8	0	876	4	176
Way	0	0	3	0	0	4	0	1	0	0	0	4	313	24
O	43	15	50	279	9	85	49	78	41	25	48	164	33	17095

FIGURE 11. Confusion matrix of XLNet.



	Area	Exp	Features	HackOrg	Idus	OffAct	Org	Purp	SamFile	SecTeam	Time	Tool	Way	O
Area	400	0	0	0	0	0	0	0	0	0	0	0	0	17
Exp	0	917	0	0	6	0	0	0	0	0	0	0	0	2
Features	0	0	250	0	0	0	0	1	0	0	0	0	0	1
HackOrg	3	0	0	1027	0	7	0	0	0	37	0	0	0	77
Idus	0	0	0	1	145	0	8	0	0	0	0	0	0	13
OffAct	0	0	0	2	0	167	0	3	0	8	0	0	0	31
Org	15	0	0	0	5	0	193	0	0	7	0	0	0	56
Purp	0	0	3	0	0	1	0	235	0	0	0	0	0	19
SamFile	0	0	0	1	0	0	0	0	683	0	0	0	0	43
SecTeam	0	0	0	14	0	3	4	0	0	491	0	0	0	31
Time	0	0	0	0	0	0	0	0	0	0	394	0	0	34
Tool	0	0	0	25	0	2	2	0	0	0	0	850	24	286
Way	0	0	2	0	0	4	0	0	0	7	0	0	323	30
O	53	7	50	243	10	82	57	90	32	29	51	160	39	15898

FIGURE 12. Confusion matrix of BERT.

E. DATA CORRELATION

This section describes in detail the conducted experiment regarding data correlation tools of ThreatWise AI. In particular, the experiment aims to assess the correlation of external as well as internal sources.

1) CORRELATION OF INTERNAL SOURCES

a: DATASET USED

Data that has been collected from the honeypot instances (i.e., internal sources). In particular, the utilised dataset includes information about cyber-attacks against the exposed services of the honeypot instances. The Dionaea honeypot attracted many attacks due to the more generic nature of the services it provides. On the contrary, cyber-attacks against the Conpot honeypot were significantly less, considering that the exposed services of Conpot are more ICS oriented which results in attracting more targeted cyber-attacks.

b: IMPLEMENTED METHOD

To prepare the input dataset for the algorithm properly, every command is separated into multiple commands based on different keywords. This approach allows the identification of frequently executed commands. In our dataset, these keywords represent SQL commands that are commonly used by attackers, such as *"EXEC"*, *"exec"*, *"DECLARE"*, *"SELECT"*, *"Drop"*, and *"Create"*. For example, the keyword *"EXEC"* is used to execute a selected procedure.

c: EXPERIMENTAL SETUP AND RESULTS

The following sequence of SQL Commands has been captured by our deployed honeypot instances [78]:

exec sp server info 1 exec sp server info 2 exec sp server info 500 select 501,NULL, 1 where 'a'='A' select 504, c.name, c.description, c.definition from master.dbo.syscharsets c, master.dbo.syscharsets c1, master.dbo.sysconfigures f where f.config=123 and f.value=c1.id and c1.csid=c.id set textsize 2147483647 set arithabort on

The sequence is split into the following five different SQL commands:

- *exec sp server info 1*
- *exec sp server info 2*
- *exec sp server info 500*
- *select 501,NULL,1 where 'a'='A'*
- *select 504,c.name, c.description, c.definition from master.dbo.syscharsets c, master.dbo.syscharsets c1, master.dbo.sysconfigures f where f.config=123 and f.value=c1.id and c1.csid=c.id set textsize 2147483647 set arithabort on*

ARL leverages a txt file which includes the mapping between MISP events and the respective entry (i.e., line number) in the alerts.json file of Wazuh which is referred to as the index. In particular, the ARL rules are associated with the index of the mapping file to identify the related MISP event ID and proceed with enriching the stored content. The enriched information includes the correlated event ID as well as the values of support, lift and confidence that have been calculated during the correlation process.

After these steps, the data was fed into the algorithm to extract different correlations between the attacks, resulting in advanced CTI knowledge. The FP-Growth [84] algorithm was used for the identification of the commands run by attackers. For example, the following rule generated by the FP-Growth shows a sequence of commands executed by attackers:

(frozenset('Drop Procedure DllUnregisterServer ', c'Drop Procedure sp_password ',c'EXEC sp_OA', 'exec DllUnregisterServer '), frozenset("Create 'WbemScripting.SWbem Locator', @objLocator OUTPUT", 'exec DllRegister Server '),

Results: By analysing the commands listed in the above rule it can be inferred that if the attacker executes the first to fourth command, there is a high probability that they will also execute the fifth and sixth commands. The correlation of the commands reveals an attacker's complete actions and in turn, leads to the enrichment of the CTI.

If: (frozenset('Drop Procedure DllUnregisterServer ',c' Drop Procedure sp_password ',c'EXEC sp_OA', 'exec DllUnregisterServer '),

Then: frozenset("Create 'WbemScripting.SWbemLocator', objLocator OUTPUT",'exec DllRegisterServer '),

2) CORRELATION OF EXTERNAL SOURCES

a: DATASET USED

To test the external sources correlation component we used a part of our collected set of documents which we used to build the domain classifier. This document collection consists of **88** web articles, all having as topic one of the domains of interest. For each article, we have identified all its relevant documents from the collection.

b: IMPLEMENTED METHOD

The criteria for assessing two articles as relevant include one of the cases where the articles describe: (i) the same cyber incident (e.g., the same attack), (ii) the same target system (e.g., vulnerabilities of a specific technology used in aviation, such as the in-flight Wi-Fi system in the aviation domain), (iii) attacks with clear similarities, for example having the same incentives and targeting the same domain, such as ransomware attacks in the naval domain, or cyber-espionage incidents against defence contractors in the navy domain. An article can also be considered relevant to another if it partly refers to the same topic. For example, an article about the vulnerabilities of two avionics systems will be considered relevant to articles that are concerned with either of those systems.

c: EXPERIMENTAL SETUP AND RESULTS

The results concerning the performance of the text retrieval module used to correlate the external sources are described in Table 11. With regard to the results, the document collection we have used here is relatively small and, consequently, many of the articles have a small number of relevant articles, according to our ground truth. Specifically, 30 of them (roughly one-third of the collection) have up to 3 relevant articles. This certainly affects the observed deteriorating results regarding the precision metric when the ranks increase, especially in the cases of $P@4$ and $P@5$. To give an example, if a document has only two relevant documents and both of them are returned in the top-2 ranks of the system's output list, which is obviously a good response from the system, $P@5$ will only be 0.4 (because the list will also contain 3 irrelevant documents). Keeping in mind the above, this module can be efficiently used to retrieve correlated textual content, as shown, for example from the results regarding P@1, P@2 and P@3.

TABLE 11. Performance of the text retrieval module, used to correlate the external sources.

P@1	P@2	P@3	P@4	P@5	MAP
0.89	0.79	0.71	0.62	0.55	0.61

VI. EVALUATION METRICS

This section provides a detailed explanation of the different evaluation metrics utilised across the experiments.

A. CYBERSECURITY CLASSIFIER OF FOCUSED CRAWLER

This section describes the evaluation metrics that have been used to evaluate the output of experiments concerning the cybersecurity classifier that is leveraged by the focused crawler.

1) RELEVANCE TO THE CS OR CTI

This metric concerns the relevance of the content to either the cybersecurity or the CTI domain (*no-csec, csec, CTI*).

2) RELEVANCE TO THE DOMAIN

This metric concerns the relevance of the content to the target domain of interest (e.g., water industry, healthcare).

3) BALANCED ACCURACY

This metric is a version of accuracy that is used when a class imbalance occurs. In our case, the class imbalance is mild, but we chose this metric to not overestimate the performance.

4) PRECISION

Precision measures how well a Retrieval System is performing in rejecting non-relevant documents.

5) RECALL

Recall measures the percentage of data samples that a model correctly identifies as relevant.

6) F1-WEIGHTED

F1-score is the harmonic mean of the precision and recall and is used to summarise the overall performance. F1-weighted is a version of the F1-score used in cases of class imbalance as it calculates metrics for each label and averages them weighted by the support of each class.

B. CORRELATION OF EXTERNAL SOURCES

The textual correlation component is in essence an Information Retrieval module. To this end, we evaluate the performance of our text similarity approach with metrics used in Information Retrieval by comparing the ground truth relevance lists to the system's ranked lists of the most similar documents, for every document in our collection. Specifically, we use the precision and **MAP** (Mean Average Precision) metrics. **MAP** is a metric used to assess the ability of a retrieval system to find many relevant documents while placing more emphasis on the higher ranks. Precision at position k ($P@k$) is the precision regarding a single query when considering the top k retrieved documents. Average Precision is defined as $\frac{1}{\mathrm{nRD}} \sum_{i=k}^{n} \frac{\mathrm{P@k}}{\mathrm{rel@k}}$, where nRD is the total number of relevant documents, n is the length of the output ranked list and $rel@k$ is a binary value that indicates whether the document retrieved at position k is relevant or not. The MAP metric is the mean Average Precision for a set of queries. The length of the output ranked lists in our real-world system is defined by the user.

C. CORRELATION OF INTERNAL SOURCES

In Association Rule Learning (ARL), different rules can be generated depending on the size of the dataset and the complexity of the commands executed by attackers. To select specific rules, various metrics can be used, such as Support, Confidence, and Lift. In general, using ARL different rules can be produced. The metrics presented can be used to evaluate the reliability and importance of the generated rules, which can help to identify patterns of attacks and improve the understanding of the attackers' behaviour.

Support, $\in \{0, 1\}$ is a metric that measures the proportion of transactions in the dataset that contain a specific itemset, which indicates how often a generated rule appears in the dataset [85].

Confidence, $\in \{0, 1\}$ is a metric that measures the reliability of a rule by showing the percentage of cases in which the consequent (Y) appears, given that the antecedent (X) has occurred. It is calculated as the number of transactions containing X and Y, divided by the number of transactions containing X [77].

Lift, $\in \{0, \infty\}$ is a metric that measures the ratio of the interdependence of observed values, meaning the ratio of observed support to expected support if X and Y were independent. If the lift is equal to one, it means the rule and the items are independent, whereas if the lift is more than one, it indicates a higher dependency [86].

D. DATA CLASSIFICATION

Standard evaluation metrics are used to assess the performance of the data classification component, namely Precision ($Prec$), Recall (Rec), F1-score ($F1$), and Accuracy (Acc). The above metrics measure the quality of predictions using combinations of True Positives (TP - the number of cases the classifier correctly predicted as belonging to the positive class), True Negatives (TN - the number of cases correctly predicted as belonging to the negative class), False Positives (FP - the number of cases falsely predicted as positives), and False Negatives (FN - the number of cases falsely predicted as negatives). In particular, **Prec:** $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$, **Rec:** $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$, **F1:** $\frac{2\times\mathrm{Prec}\times\mathrm{Rec}}{\mathrm{Prec+Rec}}$, and **Acc:** $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$.

E. NER

With regard to the comparison and assessment of the performance of the different models, the primary evaluation metrics that were employed are Precision, Recall, and F1-score for both predicted tokens and entities. Accuracy was not included as a metric since the 'O' class (i.e. not part of a named entity) is the majority and most models tend to predict it with high accuracy.

VII. DISCUSSION

ThreatWise AI framework comprises several tools that provide distinct functionalities, aiming to facilitate and enhance the capabilities of its functionality. Nevertheless, we have identified several issues during the implementation and testing of the tools. While these issues do not affect the overall quality, they have to be considered in order to improve them in future work.

An issue is encountered in the evasive crawler which provides more advanced crawling functionalities and enables the crawling of more sources, yet it lacks in terms of performance leading to the conclusion that the complexity of the crawling is proportional to the execution time of the functionality. Another issue concerns the simple correlation of the stored CTI. In particular, the correlation engine of MISP tends to correlate too many events, thus decreasing the performance of the database, and therefore the queries both via the GUI and the API might take too long to execute. Furthermore, the document collection that has been used for the training of the domain classification is relatively small, resulting in many articles having a small number of relevant articles, according to our ground truth. Apart from this, the classification of the CTI is limited since it is performed only according to the domains of interest.

Future work will include the development of tailor-made CTI according to the users' needs. In addition, for all the components that make use of textual data (for example for the cybersecurity classifier of the focused crawler, for the NER component or the text similarity module) the application of more recent and powerful algorithms such as techniques based on Large Language Models (LLM) can be tested. In addition, due to the proliferation of attacks, we plan to develop taxonomies related to cyber threat attacks and vulnerabilities. Specifically, the improved classification will enable the dynamic creation and adaptation of cybersecurity taxonomies concerning all cybersecurity domains, based on the content. NER will be used for the generation of taxonomies, alongside topic modelling techniques such as BERTopic. LLMs will also be tested in the generation of taxonomies. Lastly, in order to find meaning between the different taxonomies, ontologies will be generated using standard libraries such as OWLReady 2.[45] LLMs will also be used for the generation of ontologies and, more specifically, for finding relationships between the different taxonomies.

Concerning information gathering from social media, more platforms will be added. Finally, feature work will include the creation of ontologies. Ontologies, use taxonomies and also relations between them, which allows for the extraction of further knowledge by revealing patterns and relations in datasets that initially are not easily observed.

VIII. CONCLUSION

In this work, we have proposed a novel framework called ThreatWise AI for the gathering, analysis, enrichment and sharing of CTI. We evaluated our framework using real data collected from various online sources as well as honeypot instances deployed on the cloud. The information from all sources is stored on MISP where it is further enriched and becomes available for sharing. Concerning the enrichment, the CTI content is analysed and classified as either relevant or not to cybersecurity or CTI-domain, following a domain classification (i.e., aviation, naval, power grid).

Each component of the proposed ThreatWise AI framework provides a set of functionalities in an efficient and user-friendly manner. The results of our conducted experiments have shown that the framework's developed tools perform well even with a relatively small amount of documents. Specifically, the classification ML algorithms have shown good performance with the cybersecurity classifier introducing balanced accuracy, precision, recall and f1-weighted scores of 0.82, 0.89, 0.89, and 0.85, respectively, with a dataset of 39 annotated web articles.

While the tools developed during this research have demonstrated significant potential in addressing current cybersecurity challenges, future work will include their continuous improvement and enhancement. The aim is to ensure that these tools remain effective in the face of sophisticated cyber-attacks and continue while simultaneously improving their performance in terms of both results and resource consumption.

[45] https://owlready2.readthedocs.io/en/latest/

ACKNOWLEDGMENT

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Comput. Secur.*, vol. 72, pp. 212–233, Jan. 2018.

[2] M. Lehto, "Cyber-attacks against critical infrastructure," in *Cyber Security: Critical Infrastructure Protection*. Springer, 2022, pp. 3–42.

[3] B. Jabiyev, O. Mirzaei, A. Kharraz, and E. Kirda, "Preventing server-side request forgery attacks," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1626–1635.

[4] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A. K. Bashir, and F. A. Khan, "Securing critical infrastructures: Deep-learning-based threat detection in IIoT," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 76–82, Oct. 2021.

[5] C. Johnson, L. Badger, D. Waltermire, J. Snyder, and C. Skorupka, "Guide to cyber threat information sharing," *NIST Special Publication*, vol. 800, no. 150, 2016.

[6] T. Chantzios, P. Koloveas, S. Skiadopoulos, N. Kolokotronis, C. Tryfonopoulos, V.-G. Bilali, and D. Kavallieros, "The quest for the appropriate cyber-threat intelligence sharing platform," in *Proc. DATA*, 2019, pp. 369–376.

[7] A. Ramsdale, S. Shiaeles, and N. Kolokotronis, "A comparative analysis of cyber-threat intelligence sources, formats and languages," *Electronics*, vol. 9, no. 5, p. 824, May 2020.

[8] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, and A. Rauber, "A framework for cyber threat intelligence extraction from raw log data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3200–3209.

[9] N. Afzaliseresht, Y. Miao, S. Michalska, Q. Liu, and H. Wang, "From logs to stories: Human-centred data mining for cyber threat intelligence," *IEEE Access*, vol. 8, pp. 19089–19099, 2020.

[10] L. Spitzner, *Honeypots: Tracking Hackers*, vol. 1. Reading, MA, USA: Addison-Wesley, 2003.

[11] L. Zobal, D. Kolár, and R. Fujdiak, "Current state of honeypots and deception strategies in cybersecurity," in *Proc. 11th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2019, pp. 1–9.

[12] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A survey on honeypot software and data analysis," 2016, *arXiv:1608.06249*.

[13] A. Kyriakou and N. Sklavos, "Container-based honeypot deployment for the analysis of malicious activity," in *Proc. Global Inf. Infrastructure Netw. Symp. (GIIS)*, Oct. 2018, pp. 1–4.

[14] S. Kumar, B. Janet, and R. Eswari, "Multi platform honeypot for generation of cyber threat intelligence," in *Proc. IEEE 9th Int. Conf. Adv. Comput. (IACC)*, Dec. 2019, pp. 25–29.

[15] Z. Zhang, H. Esaki, and H. Ochiai, "Unveiling malicious activities in LAN with honeypot," in *Proc. 4th Int. Conf. Inf. Technol. (InCIT)*, Oct. 2019, pp. 179–183.

[16] S. Bistarelli, E. Bosimini, and F. Santini, "A report on the security of home connections with IoT and Docker honeypots," in *Proc. ITASEC*, vol. 2597, Jan. 2020, pp. 60–70.

[17] M. Parmar and A. Domingo, "On the use of cyber threat intelligence (CTI) in support of developing the commander's understanding of the adversary," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 1–6.

[18] C. Krasznay and G. Gyebnár, "Possibilities and limitations of cyber threat intelligence in energy systems," in *Proc. 13th Int. Conf. Cyber Conflict (CyCon)*, May 2021, pp. 171–188.

[19] J. Thom, Y. Shah, and S. Sengupta, "Correlation of cyber threat intelligence data across global honeypots," in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 0766–0772.

[20] E. Bou-Harb, W. Lucia, N. Forti, S. Weerakkody, N. Ghani, and B. Sinopoli, "Cyber meets control: A novel federated approach for resilient CPS leveraging real cyber threat intelligence," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 198–204, May 2017.

[21] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. Hoboken, NJ, USA: Wiley, 2014.

[22] P. Koloveas, T. Chantzios, C. Tryfonopoulos, and S. Skiadopoulos, "A crawler architecture for harvesting the clear, social, and dark web for IoT-related cyber-threat intelligence," in *Proc. IEEE World Congr. Services (SERVICES)*, Jul. 2019, pp. 3–8.

[23] P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, and C. Tryfonopoulos, "InTIME: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence," *Electronics*, vol. 10, no. 7, p. 818, Mar. 2021.

[24] M. Najork, "Web crawler architecture," Tech. Rep., 2009.

[25] J. M. Hsieh, S. D. Gribble, and H. M. Levy, "The architecture and implementation of an extensible web crawler," in *Proc. NSDI*, vol. 10, 2010, pp. 28–30.

[26] A. Harth, J. Umbrich, and S. Decker, "Multicrawler: A pipelined architecture for crawling and indexing semantic web data," in *Proc. 5th Int. Semantic Web Conf.*, Athens, GA, USA. Springer, Nov. 2006, pp. 258–271.

[27] F. Ahmadi-Abkenari and A. Selamat, "An architecture for a focused trend parallel web crawler with the application of clickstream analysis," *Inf. Sci.*, vol. 184, no. 1, pp. 266–281, Feb. 2012.

[28] D. L. Quoc, C. Fetzer, P. Felber, É. Rivière, V. Schiavoni, and P. Sutra, "UniCrawl: A practical geographically distributed web crawler," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun. 2015, pp. 389–396.

[29] O. Vikas, N. J. Chiluka, P. K. Ray, G. Meena, A. K. Meshram, A. Gupta, and A. Sisodia, "WebMiner-anatomy of super peer based incremental topic-specific web crawler," in *Proc. 6th Int. Conf. Netw. (ICN)*, Apr. 2007, p. 32.

[30] B. Bamba, L. Liu, J. Caverlee, V. Padliya, M. Srivatsa, T. Bansal, M. Palekar, J. Patrao, S. Li, and A. Singh, "DSphere: A source-centric approach to crawling, indexing and searching the world wide web," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 1515–1516.

[31] B. Dung Le, G. Wang, M. Nasim, and A. Babar, "Gathering cyber threat intelligence from Twitter using novelty classification," 2019, *arXiv:1907.01755*.

[32] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 3111–3119.

[34] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[36] S. Sbalchiero and M. Eder, "Topic modeling, long texts and the best number of topics. Some problems and solutions," *Qual. Quantity*, vol. 54, no. 4, pp. 1095–1108, Aug. 2020.

[37] K. Rajendra Prasad, M. Mohammed, and R. M. Noorullah, "Visual topic models for healthcare data clustering," *Evol. Intell.*, vol. 14, no. 2, pp. 545–562, Jun. 2021.

[38] R. Dzisevic and D. Šešok, "Text classification using different feature extraction approaches," in *Proc. Open Conf. Electr., Electron. Inf. Sci. (eStream)*, Apr. 2019, pp. 1–4.

[39] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," 2022, *arXiv:2203.05794*.

[40] Y. Luan and S. Lin, "Research on text classification based on CNN and LSTM," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Mar. 2019, pp. 352–355.

[41] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *Proc. IEEE 2nd Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2017, pp. 721–724.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[43] J. Zhang, Y. Li, J. Tian, and T. Li, "LSTM-CNN hybrid model for text classification," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 1675–1680.

[44] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019.

[45] S. Wang, M. Huang, and Z. Deng, "Densely connected CNN with multiscale feature attention for text classification," in *Proc. IJCAI*, Jul. 2018, pp. 4468–4474.

[46] D. Zeng, Y. Dai, F. Li, J. Wang, and A. K. Sangaiah, "Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 3971–3980, May 2019.

[47] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, "TIMiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101867.

[48] A. S. Gautam, Y. Gahlot, and P. Kamat, "Hacker forum exploit and classification for proactive cyber threat intelligence," in *Inventive Computation Technologies*. Springer, 2020, pp. 279–285.

[49] I. Keraghel, S. Morbieu, and M. Nadif, "Recent advances in named entity recognition: A comprehensive survey and comparative study," 2024, *arXiv:2401.10825*.

[50] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, and W. Pedrycz, "A comprehensive survey on applications of transformers for deep learning tasks," *Expert Syst. Appl.*, vol. 241, May 2024, Art. no. 122666.

[51] P. Evangelatos, C. Iliou, T. Mavropoulos, K. Apostolou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Named entity recognition in cyber threat intelligence using transformer-based models," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 348–353.

[52] S. R. Bowman, E. Pavlick, E. Grave, B. V. Durme, A. Wang, J. Hula, P. Xia, R. Pappagari, R. T. McCoy, R. Patel, N. Kim, I. Tenney, Y. Huang, K. Yu, S. Jin, and B. Chen. (2019). *Looking for ELMo's Friends: Sentence-level Pretraining Beyond Language Modeling*. [Online]. Available: https://openreview.net/forum?id=Bkl87h09FX

[53] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets," 2019, *arXiv:1906.05474*.

[54] X. Wang, X. Liu, S. Ao, N. Li, Z. Jiang, Z. Xu, Z. Xiong, M. Xiong, and X. Zhang, "DNRTI: A large-scale dataset for named entity recognition in threat intelligence," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1842–1848.

[55] S. Deshmukh, R. Rade, and F. Kazi, "Attacker behaviour profiling using stochastic ensemble of hidden Markov models," 2019, *arXiv:1905.11824*.

[56] Y. Wu, C. Huang, X. Zhang, and H. Zhou, "GroupTracer: Automatic attacker TTP profile extraction and group cluster in Internet of Things," *Secur. Commun. Netw.*, vol. 2020, pp. 1–14, Dec. 2020.

[57] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and Y. Kompatsiaris, "Towards a framework for detecting advanced web bots," in *Proc. 14th Int. Conf. Availability, Rel. Secur.*, Aug. 2019, pp. 1–10.

[58] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, "Detection of advanced web bots by combining web logs with mouse behavioural biometrics," *Digit. Threats, Res. Pract.*, vol. 2, no. 3, pp. 1–26, Sep. 2021.

[59] R. Broadhurst, D. Lord, D. Maxim, H. Woodford-Smith, C. Johnston, H. W. Chung, S. Carroll, H. Trivedi, and B. Sabol, "Malware trends on 'darknet' crypto-markets: Research review," Tech. Rep., 2018.

[60] R. Koch, "Hidden in the shadow: The dark web—A growing risk for military operations?" in *Proc. 11th Int. Conf. Cyber Conflict (CyCon)*, vol. 900, May 2019, pp. 1–24.

[61] C. Olston and M. Najork, "Web crawling," *Found. Trends Inf. Retr.*, vol. 4, no. 3, pp. 175–246, Jan. 2010.

[62] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: A new approach to topic-specific web resource discovery," *Comput. Netw.*, vol. 31, nos. 11–16, pp. 1623–1640, May 1999.

[63] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, Feb. 2010, pp. 441–450.

[64] P. Panagiotou, C. Iliou, K. Apostolou, T. Tsikrika, S. Vrochidis, P. Chatzimisios, and I. Kompatsiaris, "Towards selecting informative content for cyber threat intelligence," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 354–359.

[65] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[66] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1999.

[67] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser fingerprinting: A survey," *ACM Trans. Web*, vol. 14, no. 2, pp. 1–33, 2020.

[68] H. Jonker, B. Krumnow, and G. Vlot, "Fingerprint surface-based detection of web bot detectors," in *Proc. Eur. Symp. Res. Comput. Secur.* Springer, 2019, pp. 586–605.

[69] J. Soler-Company and L. Wanner, "On the role of syntactic dependencies and discourse relations for author and gender identification," *Pattern Recognit. Lett.*, vol. 105, pp. 87–95, Apr. 2018.

[70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[71] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[72] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019.

[73] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pretraining text encoders as discriminators rather than generators," 2020, *arXiv:2003.10555*.

[74] S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "MalwareTextDB: A database for annotated malware articles," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1557–1567.

[75] J. Jabez and B. Muthukumar, "Intrusion detection system (IDS): Anomaly detection using outlier detection approach," *Proc. Comput. Sci.*, vol. 48, pp. 338–346, Jan. 2015.

[76] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, pp. 1–39, Mar. 2012.

[77] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 1993, pp. 207–216.

[78] A. Papoutsis, C. Iliou, D. Kavallieros, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Host-based cyber attack pattern identification on honeypot logs using association rule learning," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2022, pp. 50–55.

[79] F. S. Tsai, "Network intrusion detection using association rules," *Int. J. Recent Trends Eng.*, vol. 2, no. 2, p. 202, 2009.

[80] C. C. Aggarwal, *Data Mining: The Textbook*, vol. 1. Springer, 2015.

[81] G. Settanni, Y. Shovgenya, F. Skopik, R. Graf, M. Wurzenberger, and R. Fiedler, "Acquiring cyber threat intelligence through security information correlation," in *Proc. 3rd IEEE Int. Conf. Cybern. (CYBCONF)*, Jun. 2017, pp. 1–7.

[82] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Quantifying the significance and relevance of cyber-security text through textual similarity and cyber-security knowledge graph," *IEEE Access*, vol. 8, pp. 177041–177052, 2020.

[83] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.

[84] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, Jan. 2004.

[85] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases*, vol. 1215, Santiago, Chile, 1994, pp. 487–499.

[86] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1997, pp. 255–264.

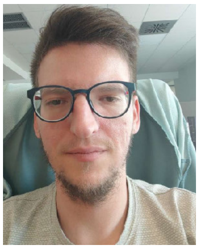
ARNOLNT SPYROS received the Diploma degree in computer engineering with specialization in software engineering from the Alexander Technological Educational Institute, Thessaloniki, in 2019, and the M.Sc. degree in cybersecurity from International Hellenic University, Thessaloniki. He is currently pursuing the Ph.D. degree specializing in IoT proactive approaches.

He joined the Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), in 2021. He has participated in more than ten European and national research projects and has co-authored several publications in the domain of CTI. His research interests include cyber threat intelligence, situational awareness, proactive security, the Internet of Things (IoT), and the Internet of Drones (IoD) cybersecurity.

ILIAS KORITSAS received the degree in rural and surveying engineering and the M.Sc. degree in geoinformatics: water resources management from the Aristotle University of Thessaloniki and the M.Sc. degree in cybersecurity from International Hellenic University. He was a Surveying Engineer and a Software Developer. He has been a Research Associate with CERTH-ITI, since 2021. He has worked on several research and development projects and has published original work in scientific journals and international conferences. His research interests include DevOps, cyberthreat intelligence, penetration testing, secure web applications development, and AI-based applications in cybersecurity.

ANGELOS PAPOUTSIS received the degree in informatics and administration from the University of Ioannina and the M.Sc. degree in informatics and information systems from Ionian University.

From 2019 to 2021, he was a Research Associate with the University of Ioannina. Since 2021, he has been a Research Associate with CERTH-ITI. Since 2019, he has worked on several research and development projects. His research interests include cybersecurity, data privacy, and artificial intelligence (AI), and more specifically the extraction of cyber threat intelligence (CTI) from different types of data.



PANOS PANAGIOTOU received the Computer and Telecommunications Engineering degree from the University of Western Macedonia and the M.Sc. degree in informatics from the Aristotle University of Thessaloniki. He has been a Research Assistant with CERTH-ITI and more specifically in the Multimodal Data Fusion and Analytics Group, since January 2019. His research interests include machine learning and artificial intelligence, with a particular focus on NLP, explainable AI, and Multimodal data applications.

DESPOINA CHATZAKOU received the degree in computer science, the M.Sc. degree in informatics and management, and the Ph.D. degree in computer science from the Aristotle University of Thessaloniki, Greece.

She has been a Postdoctoral Research Fellow with CERTH-ITI, since 2018. Since 2012, she has worked on several research and development projects and has published original work in scientific journals and international conferences. Her research activity mainly lies on artificial intelligence and in particular: web data mining, with a particular interest in data streams and social network analysis, natural language processing and text mining, modeling methods for graph and network data, and behavior analysis (e.g., sentiment analysis, affective computing, hate speech, and abuse behaviors detection), with particular focus on security applications.

DIMITRIOS KAVALLIEROS received the M.Sc. degree in ethical hacking and computer security from the University of Abertay Dundee, U.K., in 2012. He is currently pursuing the Ph.D. degree. His doctoral research is focused on IoT cybersecurity and forensics. As a Researcher, he has participated in more than 20 European-funded projects, mainly focused on cybersecurity and cybercrime (coordinating four) and he has authored more than 35 related scientific journals, conferences, and book chapter publications. His research interests include cybersecurity, cybercrime and terrorism, the IoT security, digital forensics, and cyber threat intelligence.

THEODORA TSIKRIKA received the degree in computer science from the University of Crete, Heraklion, and the M.Sc. degree in advanced methods in computer science and the Ph.D. degree in computer science from the Queen Mary University of London.

She was a Postdoctoral Researcher with CWI, Amsterdam, The Netherlands, from 2007 to 2010, the University of Applied Sciences Western Switzerland, Sierre, Switzerland, from 2011 to 2012, and the Royal School of Library and Information Science, Copenhagen, Denmark, in 2013. She has been a Postdoctoral Research Fellow with CERTH-ITI, since 2013. She has participated in more than 40 European and national research projects and has co-authored more than 100 publications in refereed journals and international conferences. Her research interests include information retrieval, data mining, and artificial intelligence (AI), and include AI-based multimodal analytics, web search, domain-specific data discovery, web and social media mining, and evaluation, with a particular focus on security and cybersecurity applications.

STEFANOS VROCHIDIS (Member, IEEE) received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Greece, the M.Sc. degree in radio frequency communication systems from the University of Southampton, and the Ph.D. degree in electronic engineering from the Queen Mary University of London, U.K.

Currently, he is a Senior Researcher (Grade C) with the Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece, and the Head of the Multimodal Data Fusion and Analytics (M4D) Group, Multimedia Knowledge and Social Media Analytics Laboratory. He has participated in more than 50 European and national projects (in more than 15 as a project coordinator, scientific or technical manager). He has edited three books and authored more than 250 related scientific journals, conferences, and book chapter publications. His research interests include multimedia analysis and retrieval, multimodal fusion, computer vision, multimodal analytics, and artificial intelligence, as well as media and arts, and environmental and security applications. He has been a member of the organization team of several conferences and workshops relevant to the aforementioned research areas.

IOANNIS KOMPATSIARIS (Senior Member, IEEE) was born in Thessaloniki, in 1973. He received the B.S. degree in electrical and computer engineering and the Ph.D. degree from the Aristotle University of Thessaloniki, in 1996 and 2001, respectively.

He is currently the Director of the Information Technologies Institute and the Head of the Multimedia Knowledge and Social Media Analytics Laboratory. He is the co-author of 178 journal articles, more than 560 conference papers, and 59 book chapters. He holds eight patents. His research interests include image and video analysis, big data and social media analytics, semantics, human–computer interfaces (AR and BCI), eHealth, and security applications. He is a Senior Member of ACM. He is a member of the National Ethics and Technoethics Committee and an Elected Member of the IEEE Image, Video, and Multidimensional Signal Processing—Technical Committee (IVMSP—TC). He has organized conferences, workshops, and summer schools. He is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING.