International Journal of Information Security (2023) 22:1511–1530
https://doi.org/10.1007/s10207-023-00703-0

REGULAR CONTRIBUTION

An autoML network traffic analyzer for cyber threat detection

Alexandros Papanikolaou[1] · Aggelos Alevizopoulos[1] · Christos Ilioudis[2] · Konstantinos Demertzis[3] · Konstantinos Rantos[3]

Accepted: 2 May 2023 / Published online: 21 May 2023

Abstract

Timely detection and effective treatment of cyber-attacks for protecting personal and sensitive data from unauthorized disclosure constitute a core demand of citizens and a legal obligation of organizations that collect and process personal data. SMEs and organizations understand their obligation to comply with GDPR and protect the personal data they have in their possession. They invest in advanced and intelligent solutions to increase their cybersecurity posture. This article introduces a ground-breaking Network Traffic Analyzer, a crucial component of the Cyber-pi project's cyber threat intelligent information sharing architecture (CTI2SA). The suggested system, built on the Lambda (λ) architecture, enhances active cybersecurity approaches for traffic analysis by combining batch and stream processing to handle massive amounts of data. The Network Traffic Analyzer's core module has an automatic model selection mechanism that selects the ML model with the highest performance among its rivals. The goal is to keep the architecture's overall threat identification capabilities functioning effectively.

Keywords Cyber threat intelligent · Cyber threat information · Information sharing · Industrial environment · Cybersecurity

1 Introduction

The fast development of new technology in recent decades has profoundly impacted human communities and the present economy [1]. The new Cyberspace [2, 3] is formed by a highly digital and linked environment that offers new opportunities and possibilities for businesses to develop extroversion-related activities and behaviors [4, 5]. Yet, there are several issues with this new cyber-ecosystem, including cyber-criminality and sophisticated persistent threats, and cyberattacks, creating an uncertain and unstable environment that undermines expected progress and prosperity [6–8]. Introducing a new generation of cyber dangers emphasizes the importance of modernizing how these difficulties are tackled [9–11]. Modern and advanced-persistence cyber threats circumvent traditional defense strategies used by enterprises that rely on the passive use of key security appliances, such as firewalls, to secure their information and anti-malware solutions [12–15]. Because of the intricacy of new threats, most successful attacks are detected only during the following forensics processes [16, 17].

Due to a lack of knowledge, non-automation, increased workload, software dependencies, the use of outdated systems, and the delayed release of critical patches, many system administrators are unable to fix all of a system's vulnerabilities quickly. The system's vulnerabilities must be completely under control if a defense is to be effective, as a successful attack could exploit just one weakness. On the other hand, network security relies heavily on the automated, intelligent gathering and correlation of suspicious actions. When combined with a broader holistic framework that considers the most recent cyber threats, this task can assist in taking

✉ Konstantinos Demertzis
kdemertzis@teiemt.gr

Alexandros Papanikolaou
a.papanikolaou@innosec.gr

Aggelos Alevizopoulos
a.alevizopoulos@innosec.gr

Christos Ilioudis
iliou@ihu.gr

Konstantinos Rantos
krantos@cs.ihu.gr

[1] Innovative Secure Technologies P.C., Thermi, Greece

[2] Department of Information and Electronic Engineering, International Hellenic University, Thermi, Greece

[3] Department of Computer Science, International Hellenic University, Kavala, Greece

appropriate measures to increase the organization's security posture [18, 19]. Even if sharing cyber threat intelligence is difficult, the benefits will be realized [20].

Using Indicators of Compromise (IOCs) to help security decision-making is a watershed moment in this process [21]. Patches, best practices in control measures, access control rules, deleting extra services, and adjusting firewall settings are all examples of IOCs, as are malware signature IDs, malicious IP addresses, malicious checksum (MD5) malware, and malicious URLs or domain names of Botnets [22–24]. In other words, this is a massive store of knowledge with tried-and-true protection tactics that is constantly updated.

Cybersecurity industrial tools could exploit currently known IOCs. Common industrial IOCs, for example, are developed by significantly using the Privileged Access Analytics (PAA) endemic to common industrial management platforms [25–27]. PAA identifies post-exploitation behaviors using stolen and abused credentials by seeing and learning how privileges are utilized throughout the company, then notifying when these are misused. Organizations are frequently at the mercy of public IOCs warning them after something has gone wrong. Industries that rely only on identifying known IOCs do not have coverage for both in-network and critical services. Strong security measures must combine the advantages of IOCs with Artificial Intelligence (AI) and Machine Learning (ML) methods to identify and stop an attacker. These state-of-the-art tools give network defenders control over their surroundings, enabling them to evade even the most skilled attackers and prevent damage before it is done [28–31].

Understanding data structures, modeling, analysis techniques, setting up data pipelines, and statistics are all part of ML practice. Data preprocessing, feature engineering, feature extraction, and feature selection may all be components of a typical ML application. Practitioners must choose an algorithm and tune hyperparameters to maximize model prediction. For example, the correct hyperparameter tuning for new workloads requires hyperparameter optimization and architecture alignment of the algorithm design. These steps may be challenging, hindering the spread of ML applications, especially in industrial cybersecurity.

This study presents a novel architecture for employing environment-appropriate IOCs based on interoperable and intelligent current ML techniques in light of the gap in applying available information based on IOCs. The proposed Network Traffic Analyzer is a sophisticated and adaptable system for monitoring and identifying security events that are damaging to an organization and is a crucial part of the CTI2SA of the Cyber-pi project. It employs advanced analysis technologies, automated management, and corrective action execution as part of a defense-in-depth and holistic cybersecurity industrial architecture, enabling real-time interaction that significantly improves the functional security features of organizations' environments.

Specifically, this paper describes the architectural and operational frameworks of the CTI2SA of the Cyber-pi project and presents one of its most critical and innovative subsystems, the network traffic analyzer. The proposed system improves active cybersecurity methods related to traffic analysis using λ architecture which can manage massive volumes of data by combining batch and stream processing techniques. The network traffic analyzer based on the autoML model selection method is proposed for the first time in the literature. It is a model selection system that determines which ML algorithm to utilize, considering numerous competing ML implementations. The system aims to simplify the adoption of the most accurate and updated ML model, capable of responding to pre-planned vulnerabilities that seek to trick the system.

Overall, the Network Traffic Analyzer is an advanced system that can analyze network traffic data in real-time and provide valuable insights into network security threats. By leveraging the Lambda architecture, the system can process and analyze large amounts of network data. The auto model selection system further enhances the system's capabilities by choosing the best-performing ML model for threat identification, improving the effectiveness of the overall architecture. This allows for faster and more accurate detection of security threats, which is critical in today's rapidly evolving cybersecurity landscape. Overall, the proposed system is a significant advancement in cybersecurity, providing improved accuracy and faster response times to emerging threats for their early detection and more efficient prevention.

The remainder of this paper is organized as follows: Sect. 2 provides context for the study approach. Section 3 is devoted to presenting the proposed architecture. The use case scenario, the suggested Network Traffic Analyzer Architecture, and its implementation are all presented in Sect. 4. Finally, part 5 brings the research to a close.

2 Related work

The fundamental disadvantage of software for analyzing network flows is that it does not provide the detailed packet-level information necessary for complete analysis. To do high-level application analysis, they do not have access to every packet in the network flow. Furthermore, the study's accuracy is affected by a fraction of the sample rate used: The sample rates supported are determined by the providers [32]. The larger the sample size, the more comprehensive the investigation. The type of sampling also affects the accuracy of the data. Furthermore, all network infrastructures must provide the protocols required for complete network traffic analysis. Similarly, when dealing with a large number of network flows, bandwidth overhead and the demand for computer resources for logical processes will substantially impact the resources required [15].

The perceived reality of the risk depiction may differ depending on their talents and expertise. Furthermore, while evaluating massive data, operators are assisted by visual means. Using relevant visualization tools as part of a complete decision-support system is an inherent and implicit requirement for demonstrating efficacy [33].

Using signatures to perform threat detection is a critical concern in these applications, notably the increasingly advanced applications that depend on Deep Packet Inspection (DPI) [34] techniques. An incomplete signature capacity may identify well-known occurrences for this signature-based malware detection, provided the proper packet is sampled, and the signature exists. Unfortunately, new harmful applications emerge that are unpredictable. Only behavioral analysis or any other progressive approach can identify these freshly released forms from innocuous files and activities [35, 36].

Hackers are constantly seeking ways to evade detection by IDS/IPS and law enforcement. For example, modern malware is systematically looking for ways to create secret connections with remote C&C servers so that hackers can propagate the damaging payload to compromised devices (bots) using hardcoded IP address pool lists. To avoid detection, malware and botnets communicate by generating the next rendezvous point with the botmasters via secret dynamic DNS services. Hundreds of random IP addresses distinguish these meeting sites. Using secret dynamic DNS services deployed on high port numbers to evade detection by IDS/IPS, botnets communicate by planning the next meeting point with the botmasters. Hundreds of random IP addresses and a Time-To-Live (TTL) for each incomplete DNS Resource Record serve to identify these meeting places. Furthermore, it is extremely challenging to locate command and control (C&C) servers and law enforcement because malware frequently employs sophisticated cryptography, as well as the Blind Proxy Redirection (BPR) technique, which repeatedly directs requests to a different group of backend servers to obfuscate traces and conceal underlying networking details [37–39]. As a result, the complexity of botnets grows [22, 40].

Demystifying Malware Traffic is the most efficient approach to cyber-attack prevention and successfully examining malware communications. Furthermore, this is the principal approach for estimating the malicious process's behavior, the goal of assaults, and the degree of degradation produced by these actions [41]. The reality is that the most sophisticated malware uses the Tor network's chaotic character [23, 24, 42] to encrypt the traces of botnets and alter the attack vectors [19, 43]. This powerful encryption-based peer-to-peer network is built on several layers, complex virtual circuits, and dynamic overlays [41, 42, 44], ensures that compromised devices and hidden services on a botnet remain anonymous. Moreover, the fact that Tor-based malware runs at the Transport layer of the OSI model complicates the investigation of this type of malware since network flow reveals clients of the Secure Socket Interface (SOCKS), which operates at the session layer [45, 46]. Consequently, the traffic generated by Tor on port 443 is similar to genuine HTTPS traffic. The study of Secure Sockets Layer (SSL) protocol changes using statistical analysis [47, 48] are among the most reliable approaches for correctly identifying Tor-generated traffic flow. For example, in a network overloaded with HTTPS traffic, statistical analysis of the associated domain name, time-to-live, and other parameters may be used to detect Tor sessions [49, 50].

On the other hand, Hsu et al. [51] suggest an anomaly detection system that examines the delay in HTTP/HTTPS client requests as a basis for a real-time botnet detection approach. Also, a completely automated fast-flux identification method based on ML and genetic algorithms with no expert input presented in the [52]. This automated technique aids in determining the uniqueness of rogue hosts' behavior from network data, even when it varies. Fast-flux detection is rendered insensitive to changes in infected hosts by the proposed method, making it more challenging for attackers to conceal their hosts as long as a representative dataset is provided. In addition, the research authors [53] used different machine learning models to identify SSH traffic with limited payload characteristics. Caglayan et al. [54] suggested an accurate ensemble method for classifying SSH traffic without relying on payload attributes. Authors of [55, 56] examine precise approaches for tracing botnets, while [57] provide a universal framework for detecting encrypted malicious communications using machine learning and [58] describe how to create a calibrated "out-of-distribution" (OOD) score based on the p-values of the relative Mahalanobis distance' to find new traffic samples. The article [59] describes an in-depth examination of HTTP, BitTorrent, and Tor traffic, as well as how to identify these protocols based on user behavior. Similarly, various research [60, 61] suggest strategies for locating Tor network encryption and relay nodes.

Previous research on network traffic analysis is devoted to external references to general techniques for attacking communication systems, particularly unused ports and services, communication channel vulnerabilities, and communication protocol vulnerabilities. Because of this, these studies do not considerably advance our understanding of network traffic flow system dangers and the seriousness of attacks against them, which frequently cause major harm and monetary losses.

In recent years, the necessity for a joint response to security crises has been stressed, and tremendous progress has been made in this area, as must be emphasized [62–64]. However, the ongoing evolution of cybercriminal tactics and technologies renders obsolete systems that do not integrate real-time information processes. Interoperability, which enables the efficient gathering, enhancement, analysis, and exchange of cyber-attack data, is also crucial. The approach of Modi et al. [65] exemplifies this type of application by proposing a layered architecture for comprehensively analyzing heterogeneous data through the interplay of its interleaves. Mantis suggests a unique approach incorporating data regarding cyber risks based on many standards. The strategy takes data from open-source data streams. However, it completely depends on internal analysis platforms, significantly limiting the generalization that should be provided in such circumstances [66]. It is a smart platform that allows threat data to be connected via an innovative, agnostic similarity algorithm. This methodology enables security analysts to connect shared patterns among ostensibly unrelated assaults, which significantly increases the system's complexity and processing resource requirements. Finally, Sengupta et al. [67] a highly complex strategy for modeling advanced persistent threat threats in a cloud computing context was proposed. The strategy is founded on game theory, in which the procedures of responding to an occurrence are modeled by minimizing the cost of security countermeasures.

Cyber-pi is an intelligent cyber threat detection and privacy protection project. The proposed CTI2SA architecture of the Cyber-pi project aims to create a layered cybersecurity architecture that can continuously upgrade vulnerability detection capabilities within an industrial environment. Regarding the specific Network Traffic Analyzer proposed for intelligent network analysis, to the best of the authors' knowledge, nothing similar has been presented in the past and the relevant literature. Therefore, we believe that its main advantage is its innovation, which is the result of combined academic and applied research. In addition, the proposed architecture provides generalization, which is one of the most important concerns in machine learning. It prevents overfitting and reduces bias and variation by employing a powerful prediction model suited to address extremely complex issues. This approach handles the dispersed, noisy misclassification points that other algorithms cannot.

Among the weaknesses of the system should be credited to the fact that it achieves a result with high accuracy but with a system characterized by a black box, which is not self-interpretable. This is particularly important as it is tough to understand how the system can make decisions. Also, due to the non-generic explainability that defines them, this system can be the target of adversarial attacks, which are particularly dangerous and quite tricky to detect.

3 CTI2SA architecture of the cyber-pi project

While following the standards of Integrated Security Information and Event Management, the suggested CTI2SA architecture (SIEM) [68] goes one step further. It offers a flexible security solution for contemporary computer systems and networks that combines various control strategies and digital security technologies. It can detect an organization's digital risks and threats using a complex collaborative framework, addressing the organization's ongoing demand for security services and crisis responses and safeguarding the crucial information it retains [20].

Particularly in response to the changing organizational structures of a contemporary, multifaceted firm, CTI2SA offers a consolidated site for analysis, alert, compliance, and reporting. It is primarily concerned with meeting each enterprise's fundamental information infrastructure needs. It has several complex techniques for keeping track of data integrity, alerting users to new dangers, locating and documenting security incidents, and quickly reacting to automated processes.

This interface provides a rapid and accurate simultaneous analysis of many security occurrences on the monitored business network while limiting the likelihood of inaccurate conclusions. The system concentrates first on the timely detection of events by automated, thorough log analysis, using a more sophisticated approach to the recommended design. Any warnings or events are shown on the system administrator's visualization console.

Data about cyber threats is acquired from trustworthy open-access sources, such as pre-built IOCs made by security professionals, and is then filtered and correlated to support infrastructure to update and enhance the predictability of CTI2SA. Component mapping enables this adjustment to the requirements of the company's business processes and information systems. Other CTI2SA subsystems may access detailed information about the open node architecture, programs, and services that could be utilized as targets or sources of malicious activity. The adaption of cyber-cognition in the STIX 2. x standard is based on comparing cyber risks with organizational characteristics [69, 70]. This approach provides a wider variety of self-healing instructions that examine the degree of compliance and aligns it with current security rules. This privacy policy production subsystem has tools for modifying and exporting SIGMA rules and analyses this. The case analysis mechanism's operational capabilities automatically include these criteria. The suggested architecture also includes a system for intelligent threat assaults that are automated.

The suggested solution uses a data-driven Network Traffic Analyzer, which employs powerful ML techniques to continually monitor incoming and outgoing network traffic to

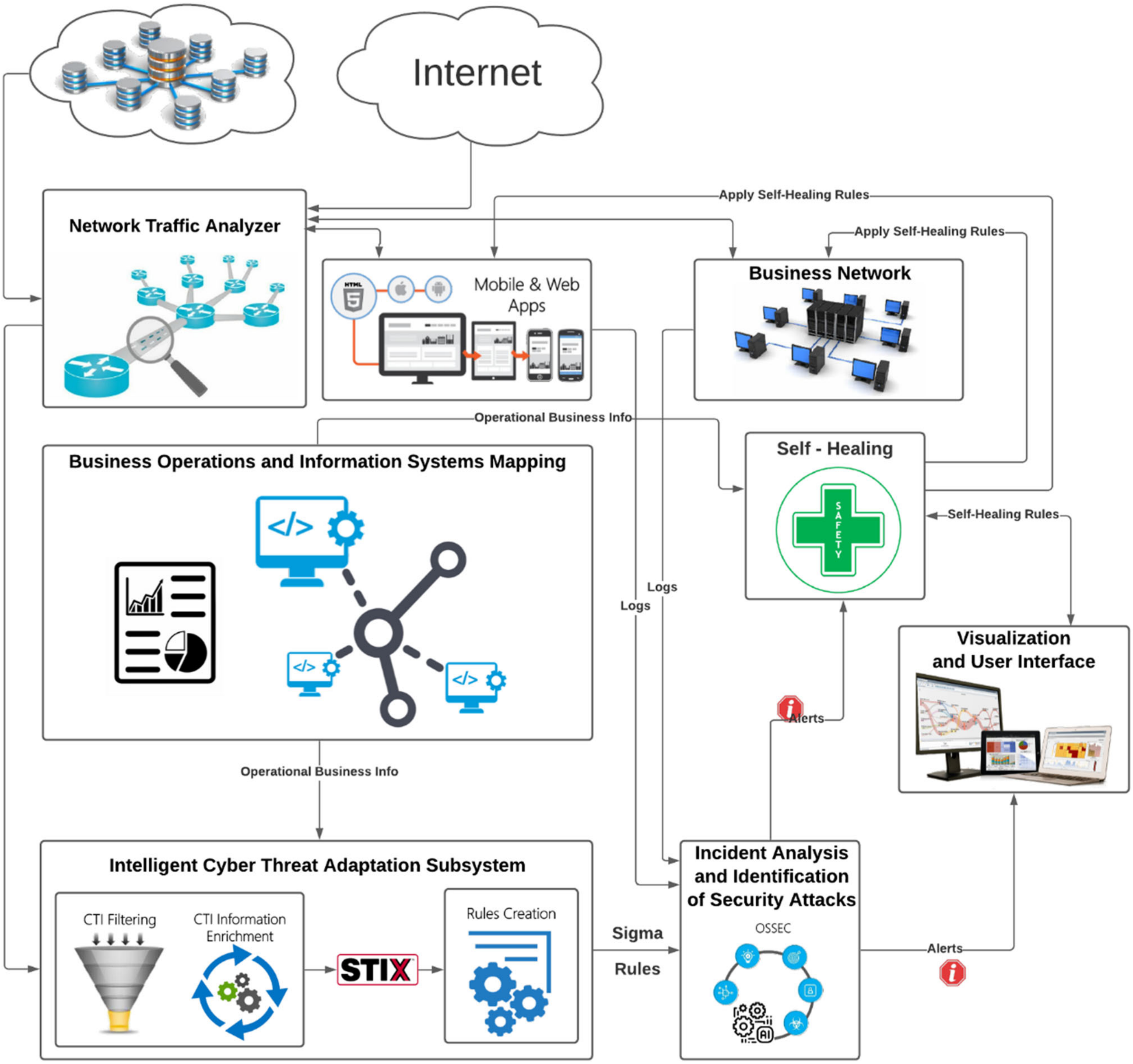
Internet
Network Traffic Analyzer
Mobile & Web Apps
HTML
Business Network
Apply Self-Healing Rules
Apply Self-Healing Rules
Operational Business Info
Business Operations and Information Systems Mapping
</>
Self - Healing
SAFETY
Self-Healing Rules
Logs
Logs
Visualization and User Interface
Alerts
Operational Business Info
Intelligent Cyber Threat Adaptation Subsystem
CTI Filtering
CTI Information Enrichment
STIX
Rules Creation
Sigma
Rules
Incident Analysis and Identification of Security Attacks
OSSEC
AI
Alerts

Fig. 1 The proposed cyber threat intelligent information sharing architecture (CTI2SA)

provide a high degree of cyber protection on supported systems and applications. Also, the proposed analyzer offers knowledge to optimize network service performance and networking resource management.

An important feature of the recommended design is the assistance provided to the visualization and interface subsystem throughout each stage of operation. This subsystem tries to efficiently represent pertinent information, allowing analysts to notice and respond to diverse events quickly. Moreover, it is critical to stress that the CTI2SA's design of preventative countermeasures is restricted to recognizing particular risks to the company and setting broad priorities. This advanced feature creates intelligent adaptation or decision-making mechanisms that guarantee smooth operation. Basic information about the parts and functions of CTI2SA is presented in Fig. 1.

Here is a detailed presentation of the CTI2SA architecture's subsystems and mechanisms.

3.1 Network traffic analyzer

The classification of network traffic [41] is typically done by specialized software [48]. Although able to analyze packet contents, some packet metrics, such as the TCP sequence and confirmation numbers, cannot be used to detect the absence of a packet or to establish their primary connections [49, 50]. This is because these tools provide only high-level information and lack alternatives that can identify events and effectively address network problems [41, 45, 71]. Similarly, firewalls offer various services such as Network Address Translation (NAT), Virtual Private Network (VPN), and traffic filtering that do not comply with network security policies [42, 47]. As they incorporate Deep Packet Inspection (DPI) functions, although they are considered a specialized, robust solution for analyzing and restricting network traffic, DPI services are particularly demanding on computing resources, as they require the ability to decrypt a Secure Socket Layer (SSL) session and subsequent retrieval of session packages must be accompanied by antivirus and instant messaging services. Also, there are many concerns about them as their use poses serious risks to users' privacy [34, 35, 42, 72].

The proposed Network Traffic Analyzer is a data-driven [47] module designed to manage and classify network traffic based on advanced automated learning techniques to overcome the above issues. It is essentially an intelligent threat detection system that acts as a gateway to the network. It is a flexible self-adapting system that automates the detection of dangerous traffic, generates warnings for further inspections, and applies the required security rules to lessen the attack surface of the organizations.

3.2 Incident analysis and identification of security attacks

In CTI2SA, the following systems perform incident analysis and intelligent identification of security incidents such as attacks.

3.2.1 OSSEC HIDS

The OSSEC Host-based Intrusion Detection System uses threat detection methods based on signatures and statistical abnormalities to assess the integrity of supervised information infrastructure files at the application and system levels [73]. It has a set of guidelines for keeping an eye on and assessing specific security incidents so that it may send out notifications. When agents are unfeasible, they can be set up to gather events from devices.

3.2.2 Decoders

Using standard and bespoke decoders with parameters that are compared with the log content for event detection, we examine the target environment's logs. The accessible collection of rules that apply the different security regulations on which the incident notifications under examination are based route any correspondences for control.

3.3 Intelligent use of CTI

Data from Cyber Threat Information (CTI) Sources are gathered and analyzed by the Intelligent Cyber Threat Adaptation Subsystem. Threats to the target network and information infrastructure are recorded and compared to the data analysis findings carried out by the supervised Information System. The Malware Information Sharing and Threat Intelligence Sharing Platform (MISP) regularly updates the intelligent cybernetics aggregation mechanism using IOCs and RSS feeds gathered from various reliable sources [74]. Each system's standards determine the accuracy of information. Filtering preserves pertinent data and compares it to the particular information infrastructure. Devices, services, item IDs, IP addresses, geolocation data, and dependencies are among the features. Consequently, custom knowledge is produced in STIX 2. x files, which match the fields required to draught SIGMA rules.

3.4 Business operations and information systems mapping

The organization's characteristics are explained through the mapping process, and the technological environment is shown on the system console. The latter makes it easier to manually record company assets, including network services, nodes, and communications. The Dependency Mapper program specifically implements the basic operations of the system. A graphical data management interface on the depmapper represents a deployment model. The ability to export graphs to picture files (jpg, png) and for data exchange has improved thanks to the dep mapper's adaptation to the mapping technique (JSON). In reality, creating image files make it easier for users of third-party applications to share pictures and connect with stakeholders. Creating JSON-formatted files further enables the inclusion of graphs to the depmapper for a later examination.

The layout model specifically shows the components, hardware components (nodes), and connections between them. The depmapper may also merge nodes, add tags, and offer many independent descriptions for each node. Information on the operation's target, available software services, geographical location, structural and procedural dependencies, data dependencies, and risk level are all included in the descriptions. The significance or sensitivity of the available services and data determines the risk level. Inside the organization, user surveys are filled out to collect this data. The data are gathered by an analyzer and entered into a depmapper, which displays text and graphs in JSON format for use by other subsystems.

3.5 Visualization and user interface

The creation of diagrams, graphs, and other visual information is related to the visualization of security data obtained from various log sources. The data structure is organized and categorized by the display approach. The sections of this subsystem include user interface and data visualization. A notice either requires immediate action or is merely informative. Categorizing the various sorts of alerts is beneficial and permits the development of a monitoring strategy [75]. According to the level of completion of the necessary study undertaken by managers and analysts, notifications are categorized.

An administrator can instantly change an alert's content, edit it later, or assign the processing of the alert to an analyst via ticket requests. As a result, the state of notifications is fresh, ongoing, and finished. The presentation approach also makes all of the data accessible to users. Self-healing system operations can be approved or activated via the user interface method. It is a fully programmable environment, which makes it possible to gather crucial data and react to occurrences quickly. It also enables the observation of historical events and the analysis of important statistics. Mobile devices and web browsers can access services thanks to the interface's support for web environments.

3.6 Self-healing policies

Self-Healing Policies are formed from the organization's decision-making and prioritization processes and are transparently and interoperably recorded in the system database. Threats, Policies, and Self-Healing Rules are contained in the database [70]. The Threat panel consists of the threat id, type, and threat group fields. A technical Command Line Interface (CLI) format and a generic format, understandable by humans, are used to store self-healing commands. Included in the self-healing policy is entries for the CLI commands that pertain to the central nodes. CLI instructions are appropriately synthesized for execution on network-end devices, such as routers, switches, firewalls, agents, and AV software. Preventing a danger can be accomplished by halting network activity in a timely manner or rendering an attack-related device unreachable.

Particularly, Self-Healing instructions offer three adjustable execution options via the system's console:

1. Inform the administrator of the necessary actions to avert a hazard or reduce the risk (recommendations).
2. Execution after administrator permission.
3. Execution automation, assuming the administrator has selected the desired configuration.

Via control command flows, this subsystem gets data from the OSSEC system and the organization's business operations and information systems mapping module. It communicates back and forth with the Visualization and User Interface subsystem through data streams. The administrator receives self-healing instructions, requiring authorization to execute a task. The administrator's selection is then sent to the self-healing subsystem. The instruction is immediately carried out if the administrator accepts. The self-healing command is sent to the administrator as a suggestion if the activity is denied. Asynchronous requests and responses are exchanged in prior communication.

When an incident is identified, the Decision Engine of the Self-Healing Module chooses which policy should be used. If a value in the Threat Category column in the Policies database matches, the procedure involves executing a command. The Threat Group field, which is more inclusive, is checked in its place if the Threat Type field is left blank. The Visualization and User Interface subsystem receive the event and transmits it, where it displays the pertinent breach alerts. The Secure Shell (SSH) protocol is typically used to remotely apply the Self-Healing rules to the nodes while recording the specifics of how an alternate command was executed in a log file.

CTI2SA demonstrates comprehensive network monitoring characteristics and effectively extends security concerns resolution to different levels (systems, services). The technology integrates aid in reducing the complexity of modern assault tactics. This is accomplished by installing specialist security software, whose primary responsibilities satisfy the requirement for regular checks to identify risks, update security rules, and maintain an acceptable security posture for the organization.

4 Use case of network traffic analyzer

To completely understand the network environment and any potential threats, professional analytic services are required due to the growing requirement for security incident management. When supplemented with data from the global threat landscape, this information enables an organization to respond to occurrences that may harm it with knowledge and precision. A vital tool in this direction is the categorization of network traffic, which is a practical approach to the design, management, and surveillance of networks and the detection of attacks or the study of cybercrime. In particular, the recognition and categorization of encrypted traffic are considered essential processes of operational security and shielding network applications. However, it is one of the most severe challenges of modern computing.

In particular, encryption gives security and privacy to users by concealing the flow of data and preventing their identity

Table 1 Darknet network traffic details

ID	Traffic category	Applications used
0	Audio-stream	Vimeo and YouTube
1	Audio-stream	Crypto streaming platform
2	Browsing	Firefox and Chrome
3	Chat	ICQ, AIM, Skype, Facebook, and Hangouts
4	Email	SMTPS, POP3S and IMAPS
5	P2P	uTorrent and Transmission (BitTorrent)
6	File transfer	Skype, SFTP, FTPS using FileZilla and an external service
7	File transfer	Crypto transferring platform
8	Video-stream	Vimeo and YouTube
9	Video-stream	Crypto streaming platform
10	VOIP	Facebook, Skype, and Hangouts voice calls

[76, 77]. Moreover, it makes it difficult for network analysts to identify and classify critical business applications, impedes the quick prioritization of high-priority operations from reaching optimal performance, respectively, the rapid increase in the use of advanced methods of encryption of web traffic has changed the landscape of the threat since cybercriminals use them to secure their malicious activities.

The pertinent literature contains information on the dataset and evaluation. The pertinent literature contains information on the dataset and evaluation. One well-known example is Tor (The Onion Router), which is frequently used to spread the latest, most sophisticated generations of malware. The dataset used in this study was called CICDarknet2020, and it contains both darknet traffic as well as matching regular traffic from Audio-Stream, Browsing, Chat, Email, P2P, Transfer, Video-Stream, VOIP, Files, Session, and Authentication, regardless of whether Tor and VPN infrastructure was being used [78]. Table 1 outlines the categories utilized and the apps that use them.

In the proposed approach, the scenario is a multi-classification problem that tris to identify and classify Tor or VPN-encrypted traffic. The dataset has a total of 141,534 data (feature vector) samples, with 93,357 samples (classified as a non-Tor class), 1393 samples (classified as a Tor class), 22,920 samples (classified as a VPN class), and 23,864 samples (classified as a non-VPN class). The whole dataset splitted in two individual parts (70% training and 30% testing): training set (65,390 non-Tor, 999 Tor, 15,993 VPN and 16,692 non-VPN samples) and testing set (27,967 non-Tor, 394 Tor, 6927 VPN and 7172 non-VPN samples).

The functioning of basic network protocols and the acknowledgment mechanism for safe data submission and reception formed the basis of the network traffic analysis and feature extraction methodology. In particular, lower-layer transmission data and preprocessed network transaction data were used. Each distinct sample has a flow-id, a class, and 80 characteristics, some of which are as follows: Several more network traffic data include the following: source IP address, source port, destination IP address, destination port, internet protocol version, timestamp, duration, the total number of packets from source to destination [78].

As a method for validating test datasets, cross-validation is employed. A technique for determining how effectively the results of statistical research may be applied to another data set is cross-validation. It is a resampling technique that analyses and trains a model using several rounds and varied data subsets. When a user wants to judge how well a predictive model will perform in the actual world and the aim is prediction, this method is most frequently used. A model is typically given access to two datasets in a prediction challenge: one with available data for training (training dataset) and the other with unknown data for evaluation (or first-seen data) (testing set). To evaluate a model's ability to predict data not included in its estimation, identify faults like overfitting or selection bias, and determine how the model will generalize to various datasets, cross-validation is performed.

Network traffic analysis and classification is a complete security solution that supplements and expands the capabilities of log analysis tools and endpoint detection and response solutions. Network traffic classification enables several network security or quality of service (QoS) features (for instance, depending on port number or protocol). It is the ideal place to start for a more proactive security posture since it provides immediate benefits and is often easier to adopt and administer than other solutions. It should be made clear that classifying network traffic allows for grouping traffic (i.e., packets) into traffic classes or groups depending on whether the traffic fits specific requirements.

There are three basic approaches for classifying network traffic: port-based, payload-based, and ML-based [41, 44]. Specifically [46, 79]:

1. Port numbers can be used to identify services that are capable of and have been assigned to handle specific network traffic. By processing the port number found on the packet's header, the system can classify the data and map them to a specific service [48].
2. In the payload-based methods, the classifier is aware of the structure of each application's packet payload. These methods, also known as Deep Packet Inspection (DPI), examine the contents of packets by referring to network application signatures in the traffic. Most payload-based methods scan the packet's contents to identify malicious content signatures retrieved from dedicated databases

and exchanged as CTI. When compared to other procedures, this yields more precise findings [34, 72]. The DPI approaches have high computational costs, need greater processing demand on identifying devices, and the implementation in encrypted communication is difficult or impossible. Finally, because the contents of a packet are viewed, are violates privacy laws and regulations.

3. ML-based strategies can overcome port and payload-based systems' constraints [58, 79]. The ML detection models are trained using a corpus of correctly labeled samples (containing both normal and malicious network traffic samples). These trained ML detection models are then used to detect malicious network traffic. Using ML approaches for traffic categorization decreases computing costs and allows for the rapid identification of encrypted communication.

As it is understood, the analysis and categorization of network traffic are an urgent need for the security of information systems. This analysis is usually performed through special software applications. A serious disadvantage of these applications is related to the need to restructure messages and entities at higher levels, with added complexity, computing resource requirements, and the production of high rates of false alarms. A significant development in addressing the above disadvantages of traditional systems is the automated ML (AutoML) solutions for real-time monitoring, analysis, and categorization of network traffic.

The success of ML is critically dependent on the performance of particular complex tasks by human-ML professionals. Using the appropriate data preprocessing, feature engineering, feature extraction, and feature selection techniques may be necessary for a typical ML application. To increase the prediction performance of their model, practitioners must choose an algorithm and tweak hyperparameters after these steps. These stages may be difficult, posing significant barriers to utilizing ML. For example, the performance of a given technique is determined by both the underlying quality of the algorithm and the specifics of its tuning, and it can be challenging to determine whether a particular method is better or better adjusted. To address the problem, one idea is to tune all ML algorithms with the same hyperparameter optimization toolbox and publish the results. Similarly, the accurate hyperparameter optimization of baselines can be enhanced over the most recent state-of-the-art results and newly presented methodologies. Approaches for automating the time-consuming, error-prone process of adjusting hyperparameters to new workloads include algorithm configuration and hyperparameter optimization.

By facilitating the creation of straightforward, consistent interfaces to multiple ML algorithms, AutoML offers techniques and procedures to increase ML efficiency and accelerate research (all state-of-the-art ML algorithms). Specifically, autoML fully automates intelligent algorithm implementation tasks, involving every step of the individual processes from data preprocessing to final model development. It is a complete solution of a high degree of automation from end to end, offering significant advantages in producing more straightforward solutions faster creation without the requirement of human supervision.

This work proposes the creation of an innovative λ-Architecture Network Traffic Analyzer, which, significantly improves the mechanisms of active security of CTI2SA. The λ-architecture [80] is a data-processing architecture meant to manage vast amounts of data by utilizing both batch and stream processing approaches [81]. It uses batch processing to offer detailed and accurate views of batch data and real-time stream processing to provide live data views to balance latency, throughput, and fault tolerance [82]. Specifically [83–85]:

1. In the batch layer, raw data are indexed so that end users may query and examine all past data. Because batch indexing takes some time, a big data window is usually momentarily inaccessible for end users to analyze.
2. The temporal window of unanalyzable data is decreased by the real-time or speed layer, which employs stream processing technologies to quickly index recent data currently unavailable for querying in the batch/serving levels. This facilitates lowering the batch/serving layers' intrinsic latency (i.e., the time it takes to make data available for analysis).

A depiction of the λ-architecture is presented in the following Fig. 2.

The suggested method prevents distributed systems' common incidence of data inconsistency. In a distributed database, data inconsistency is possible because node or network failures may prevent data from being delivered to all copies. Put another way, one copy of the data can have the most current value while the other still has the original value. The indexing procedure may ensure that the data in both the batch and speed layers represent the most current state since data in the λ-Architecture are processed sequentially (rather than in parallel with overlap, as may be the case with activities on a distributed database).

This design is built on distributed, scale-out technologies that can be expanded by adding more nodes, but it does not define which technologies should be used. This is feasible. As a result, the λ-Architecture may be handled apart from the data. t the speed, batch, serving, and data source layers.

As previously mentioned, the λ-Architecture is built on distributed systems that provide fault tolerance; as a result, in the event of a hardware failure, other nodes are available to complete the work. Furthermore, because all data are

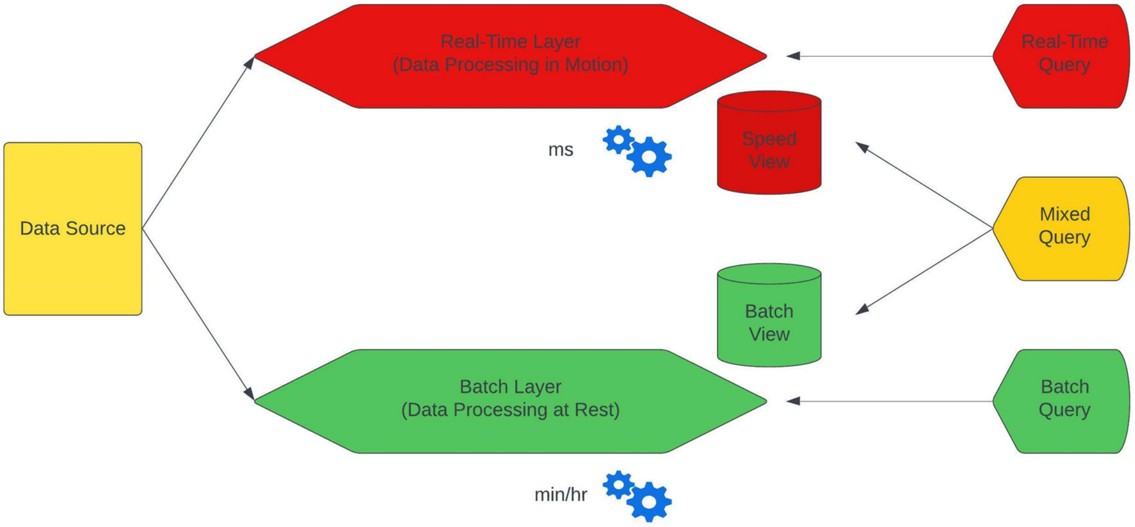

Fig. 2 λ-Architecture

kept in the batch layer, any indexing errors in the serving or speed levels may be fixed by restarting the indexing process at the batch/serving layers and letting the speed layer continue indexing the most recent data.

All indexes can be created from this data collection since raw data are preserved for indexing, serving as a record system for data that can be studied. The indexing code may be modified and restarted to reindex all data if there are any bugs or omissions.

The λ-Architecture is also built on a data model that includes an append-only, immutable data source that acts as the system of record. Instead of replacing current events, it is intended to accept and manage timestamped events tied to them. The natural time-based ordering of the data determines the state. It is designed to cope with immutable data sets that increase over time, which is the nature of the security events generated by the probes, searching through vast amounts of data kept in large repositories for trends or unusual patterns. For the reasons above, it is the optimal design for large-scale cybersecurity applications [80, 86].

The λ-Architecture Network Traffic Analyzer is proposed for the first time in the literature. It is an autoML system that optimally combines a batch engine to train the ML model with historical data. Specifically, it is an autoML model selection system that chooses which ML algorithm to use, including multiple competing ML implementations for the purpose of decision-making under uncertainty.

AutoML is a method for automating some of the more complicated or innocuous tasks in the machine learning lifecycle [87, 88]. This allows persons with no academic or practical training in ML to engage in AI development. The followings are the most notable advantages of autoML [89, 90]:

1. Efficiency: autoML aids users in transferring data to training algorithms and locating the optimal neural network design for a particular task. Using autoML, tasks that normally require hours may frequently be performed in minutes. This saves data scientists a great deal of time.
2. Scalability: autoML contributes to the democratization of ML by making machine learning techniques and technology accessible to unskilled users. AutoML technologies allow businesses to expand their AI implementations by overcoming the talent gap.
3. Error Correction: Before autoML, data scientists were required to perform tedious, manual procedures on their data. These labor-intensive procedures frequently resulted in a human mistake. AutoML made it possible for data scientists to reduce or eliminate the time-consuming, repetitive manual processes.

One of the most valuable characteristics of the method is hyperparameter optimization. It is a tuning of choosing a set of optimal hyperparameters for a learning algorithm [89, 91]. A hyperparameter is a parameter whose value controls the learning process. The importance of other parameters is learned.

Bayesian optimization is used to tune the hyperparameters. A probabilistic surrogate model and an acquisition function that chooses which point to look at next are the two main parts of the iterative Bayesian optimization method. The surrogate model is adjusted to all prior observations of the target function in each cycle. The usefulness of various prospective places is then determined by the acquisition function by balancing exploration and exploitation using the prediction distribution of the probabilistic model. Comparatively cheap and capable of being completely optimized, assessing the acquisition function can be done instead of the pricey evaluation of the black box function. Although many acquisition functions exist, the expected improvement (EI) [92]:

$$\mathbb{E}[\mathbb{I}(\lambda)] = \mathbb{E}\big[\mathbf{max}\big(f_{\mathbf{min}} - y, 0\big)\big]$$

is a popular option because, assuming the model prediction y at configuration follows λ a normal distribution, it can be calculated in closed form:

$$\mathbb{E}[\mathbb{I}(\boldsymbol{\lambda})] = \left(f_{min} - \mu(\boldsymbol{\lambda})\right)\Phi\left(\frac{f_{min} - \mu(\boldsymbol{\lambda})}{\sigma}\right) + \sigma\phi\left(\frac{f_{min} - \mu(\boldsymbol{\lambda})}{\sigma}\right)$$

where $\varphi(\cdot)$ and $\Phi(\cdot)$ are the standard normal density and standard normal distribution function, and f_{min} is the best observed value so far.

A corresponding real-time ML engine uses a timely autoML model that periodically updates the vulnerabilities identification ability. The purpose of this system is to enable the easy adoption of the most accurate ML model that can analyze network traffic [48] and, at the same time, respond to premeditated vulnerabilities which seek to deceive the system. A graphical depiction of the λ-Architecture Network Traffic Analyzer is shown in the following Fig. 3.

In particular, the initial stage of the proposed model's operation anticipates the extraction of the required features from each data stream's network traffic. These data are then saved in the historical data storage and utilized to train the ML model [93, 94].

A hybrid automated IP flow analysis technique was used to collect the data, whose basic modelling concept is based on the open-source framework Stream4Flow [95, 96]. In particular, Stream4Flow provides a comprehensive solution for IP flow analysis; it is possible to connect to most IP flow network detectors and integrate tools for data collecting, processing, manipulation, storage, and display [97]. Due to the framework's scalability, it is appropriate for processing network traffic in a wide variety of heterogeneous networks with scalable capabilities. Its distributed structure enables large-scale analyses that are computationally intensive. The method gives IP flow analysis findings with a few-second latency, which enables real-time investigation of suspicious situations [44].

The IPFIXCol collector (https://github.com/CESNET/ipfixcol2) serves as the implementation concept's foundation. Figure 4 depicts a general overview of its architecture. IPFIXcol is a technology for intricately processing IP streams from diverse sources. This adaptable flow collector enables the translation of incoming IP stream data to JSON format and supports all widely used network protocols. The collection core supports input, intermediate, and output plugins for acquiring, handling, and archiving streaming data.

The network traffic analyzer was added to the framework in question to extract the key characteristics that can define the nature of the data included in network traffic. Specifically, the open-source CICFlowMeter framework (https://github.com/ahlashkari/CICFlowMeter) was integrated as a plugin to the intermediate API.

This application uses CICFlowMeter to evaluate bidirectional network traffic flow and extract statistical characteristics and flow data. It outputs a predetermined list of features, but new features, such as fine-tuning the timeout duration of a flow, can be added on a case-by-case basis (TCP flows are usually terminated during the connection when the FIN packet is received, while UDP streams are completed with a stream timeout).

The suggested design is shown in Fig. 5 when the intermediate API of the original Stream4Flow architecture is extended by the CICFlowMeter framework.

More than eighty network traffic analysis elements may be derived from the framework's output, which includes six labeled columns for each flow (FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort, and Protocol). The flow timeout value in CICFlowMeter can be chosen randomly using a case-specific approach (e.g., 600 s for both TCP and UDP).

Just autoML is used for training, and the best hyperparameters from the winning algorithm are provided to the real-time engine to control network traffic. The training procedure is periodically repeated exactly when the data in the historical data storage increase by 30%. After that, the winning algorithm is once more submitted to the real-time engine for hot route control. The system is retrained using cross-validation processing and all possible ML techniques. The ten methods with the best classification accuracy are presented in Table 1 below, which illustrates a typical use of the autoML mode used in Network Traffic Analyzer.

Also, the Figs. 3, 4 and 5 depict the performance metrics of the best model Light Gradient Boosting Machine. After training the models with the AutoML method, the system demonstrates the performance metrics that are presented in Table 2 by Light Gradient Boosting Machine (LGBM), Random Forest Classifier (RFC), Extra Trees Classifier (ETC), Decision Tree Classifier (DTC), Gradient Boosting Classifier (GBC), k-Neighbors Classifier (k-NC), Linear Discriminant Analysis (LDA), Ridge Classifier (RC), Ada Boost Classifier (ABC), Quadratic Discriminant Analysis (QDA).

Using a set of assessment measures that show how well each model performed on the test dataset lustrates the quality and accuracy of the compared models. It must be highlighted

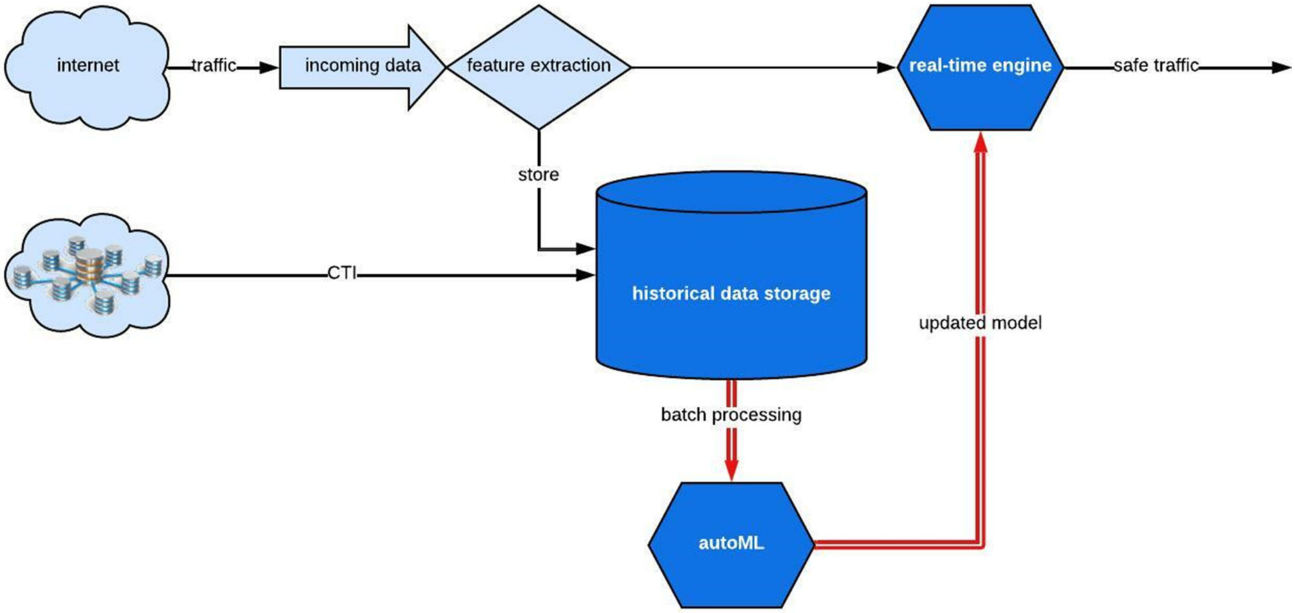

Fig. 3 The proposed network traffic analyzer architecture

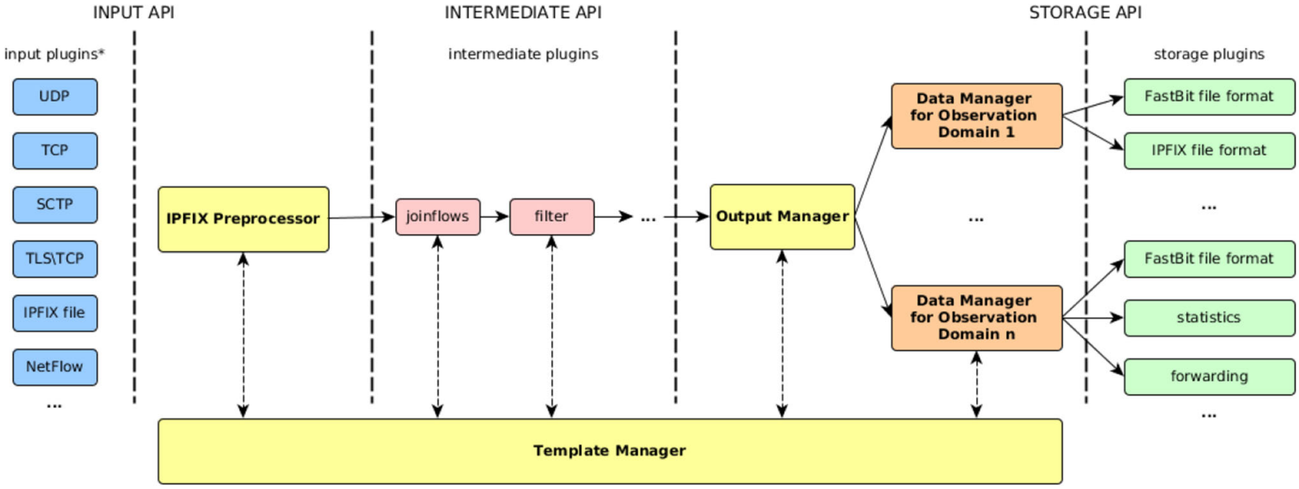

Fig. 4 The IPFIXcol architecture (https://stream4flow.ics.muni.cz/)

that the suggested method exclusively employs the cross-validation data-split method in all training scenarios. The table above displays the following statistics:

1. Accuracy: The proportion of correct classification predictions made by the model.
2. AUC: The area under the receiver operating characteristic (ROC) curve is the AUC. This scales from 0 to 1, with a more significant number indicating a higher-quality model.
3. Recall: The percentage of rows with this label that the model predicted correctly. Also known as the “true positive rate.”
4. Precision: The percentage of correct positive predictions made by the model. (Positive predictions are the sum of false positives and genuine positives.)
5. $F1$: The harmonic mean of precision and recall is used to get the $F1$-score. $F1$ is a useful metric when there is an uneven class distribution, and you want to strike a compromise between precision and recall.
6. Kappa: Cohen’s kappa coefficient (k) is a metric used to determine the inter-rater reliability of qualitative items. It is typically a more reliable measure than simple agreement estimates, as it considers the probability of the agreement occurring by chance.
7. MCC: It is a measure of association between two variables used to assess categorization quality. It considers

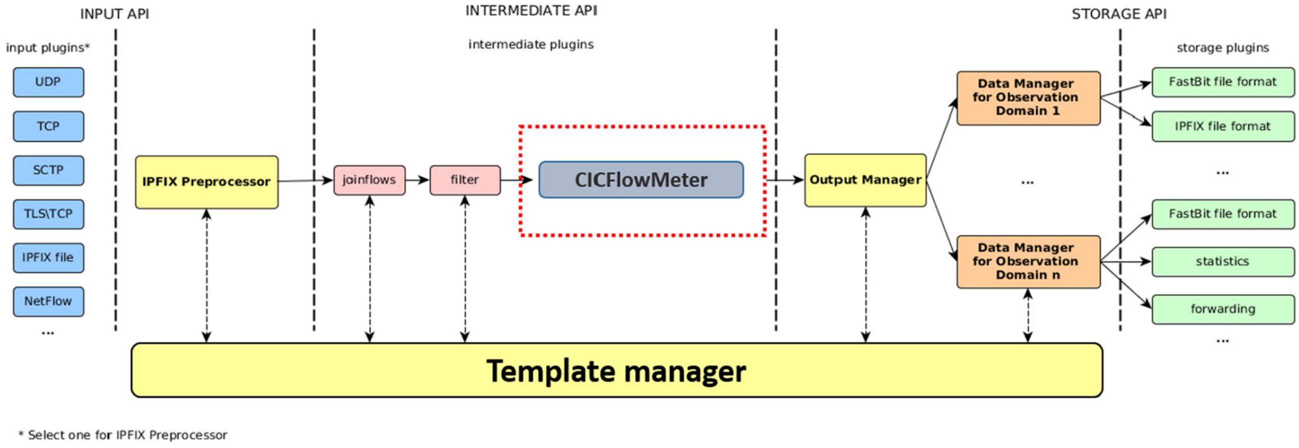

Fig. 5 Graphical illustration of the architecture of IPFIXcol with CICFlowMeter plugin

Table 2 Performance metrics of the autoML method (model selection process)

Model	Accuracy	AUC	Recall	Prec	$F1$	Kappa	MCC	TT (Sec)
LGBM	0.9896	0.9997	0.9670	0.9896	0.9896	0.9796	0.9796	10.321
RFC	0.9880	0.9994	0.9581	0.9880	0.9880	0.9765	0.9765	22.577
ETC	0.9875	0.9992	0.9628	0.9875	0.9875	0.9755	0.9755	16.963
DTC	0.9842	0.9902	0.9570	0.9842	0.9842	0.9690	0.9690	3.620
GBC	0.9767	0.9989	0.9397	0.9768	0.9766	0.9542	0.9542	317.461
k-NC	0.9446	0.9823	0.8133	0.9433	0.9435	0.8899	0.8901	19.832
LDA	0.9231	0.9854	0.8510	0.9238	0.9233	0.8494	0.8495	4.570
RC	0.9203	0.0000	0.8307	0.9192	0.9195	0.8424	0.8426	0.280
ABC	0.8525	0.9675	0.7639	0.8770	0.8581	0.7204	0.7262	16.659
QDA	0.8258	0.9662	0.7558	0.8523	0.8151	0.6703	0.6871	2.617

genuine and false positives and false negatives, and it is often a balanced metric that may be applied even when the classes are substantially varied in size. Its interpretation is comparable to the Pearson correlation coefficient because it is a correlation coefficient between observed and anticipated classes, returning a number between 1 and + 1.

8. TT(sec): Time to train the model.

In addition to the metrics listed above, the AutoML technique gives three more ways to understand the classification model: the confusion matrix, ROC curves, and learning and validation graphs.

The confusion matrix (Fig. 5) helps determine where misclassifications occur (where classes become "confused" with one another). Each row represents the actual value for a given label, and each column contains the predicted labels generated by the model. The micro-averaged accuracy is calculated by adding the number of true positives (TP) and true negatives (TN) for each possible value in the target column and dividing it by the number of TP and TN for each potential value (Fig. 6).

ROC (receiver operating characteristic) curves are graphs demonstrating the performance of a classification model's overall categorization thresholds. It displays the diagnostic performance of a classifier system as its discriminating threshold is altered. This curve depicts the True Positive Rate (also known as recall) and False Positive Rate. It lowers the threshold for categorization leading more objects to categorized as positive, increasing both False Positives and True Positives.

Figure 7 depicts the high-performance of the proposed model. The curve is in the up-left corner, which means the model has a perfect separability measure because it can perfectly distinguish between classes.

The validation and learning curves (Fig. 8) display the validation and training scores for increasing amounts of training samples. It is a method for determining the benefits of increasing the amount of training data and if the classifier suffers more from a variance or bias error.

It must be noted that the hyperparameter tuning selects the best set of hyperparameters for a machine learning algorithm to improve its performance on a specific task. One common

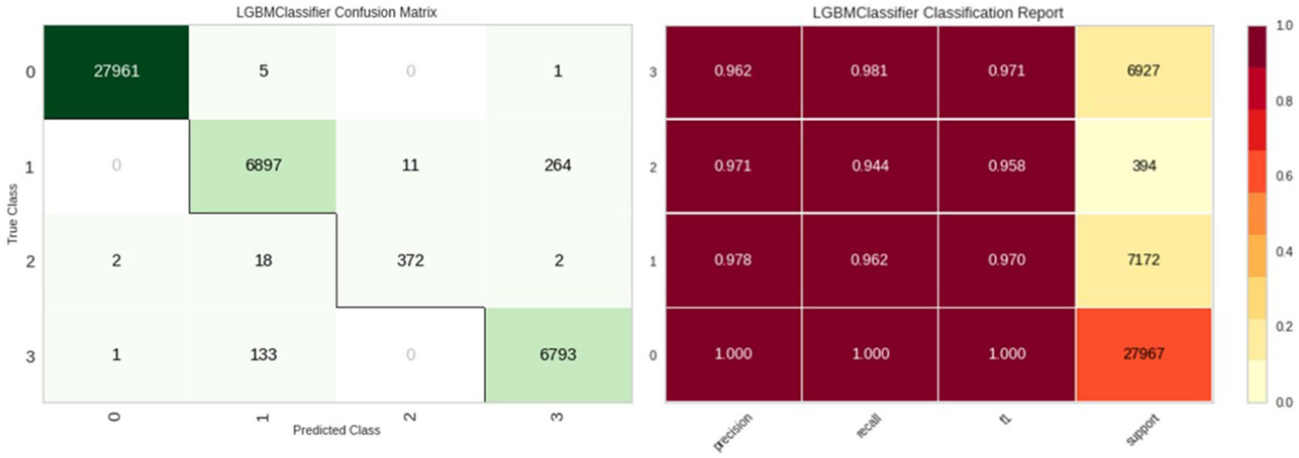

Fig. 6 Confusion matrix and Precision-Recall-$F1$-score of the light gradient boosting machine model

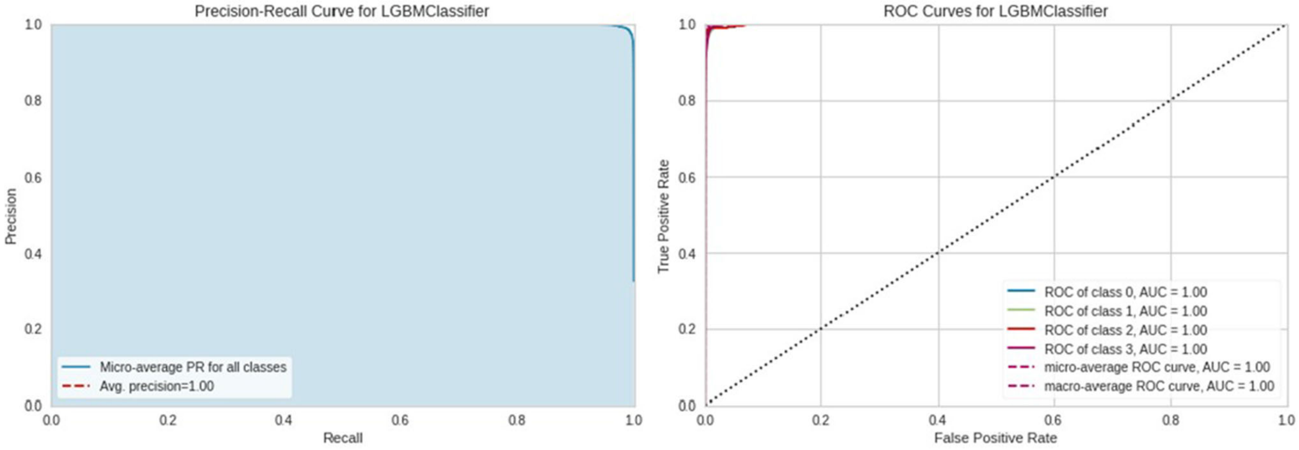

Fig. 7 Precision-recall and ROC curve of the light gradient boosting machine model

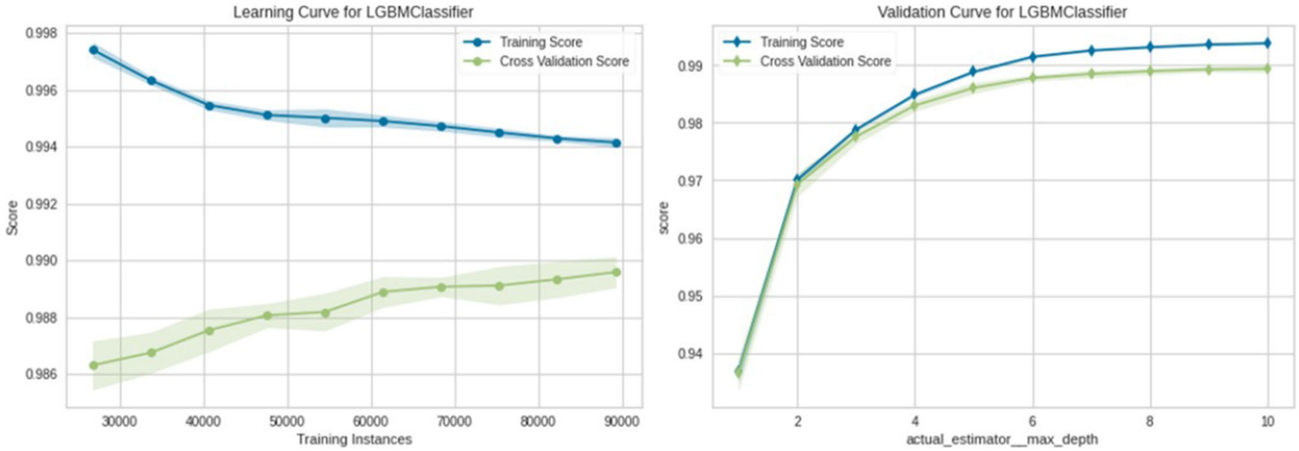

Fig. 8 Learning and validation curve of the light gradient boosting machine model

approach to hyperparameter tuning is to use learning and validation curves.

A learning curve displays a model's performance on training data as a function of the training data. When a model performs well on training data but badly on test data, it may be used to determine if it is overfitting or underfitting (performing poorly on both the training and test data).

A validation curve, on the other hand, shows the performance of a model on the validation data as a function of a hyperparameter. It can help identify the best value of the hyperparameter that maximizes the model's performance on the validation data.

Combining the information from the learning and validation curves allows us to identify the best set of hyperparameters for a given model and task. Specifically, we can identify the hyperparameters that lead to the best performance on the validation data while avoiding overfitting or underfitting.

To tune hyperparameters using learning and validation curves, we first train the model with different sets of hyperparameters and plot the learning and validation curves for each set. Then, we compare the curves and identify the set of hyperparameters that lead to the best validation performance without overfitting. This set of hyperparameters can then be used to train the final model for deployment.

While it is true that many existing Deep Learning solutions [98–101] can handle data complexity, the proposed contribution of the Network Traffic Analyzer may outperform them due to several reasons, such as:

1. Hybrid architecture: The proposed system uses a hybrid architecture that combines batch and stream processing to handle large amounts of data. This architecture allows for real-time processing of data, which is essential in detecting and mitigating cyber-attacks in industrial environments.
2. Auto model selection: The core module of the Network Traffic Analyzer uses an auto model selection system to choose the best-performing Machine Learning model among competitors. This feature ensures that the system always uses the best possible model, which can lead to better performance and accuracy.
3. Integration with other cybersecurity tools: The proposed system can be integrated with other cybersecurity tools, making it a comprehensive solution for detecting and mitigating cyber-attacks in industrial environments. This integration can lead to better overall performance and more effective protection against cyber threats.
4. Tailored for the specific use case: The proposed system is tailored for the specific use case of detecting and mitigating cyber-attacks in industrial environments. This specificity allows the system to optimize its performance and accuracy for this use case, leading to better results than a more generalized solution.

Also, it must be noted that the proposed architecture was proposed for the specific use case, considering the nature of the data, the computational resources available, and the desired performance metrics [102–106]. To explain why this architecture is the optimal solution for the industrial environment, one can provide several justifications, including:

1. Scalability: The Lambda architecture used in the proposed system is scalable and can handle large amounts of data, making it suitable for industrial environments with a high volume of data to process.
2. Real-time processing: The combination of batch and stream processing in the Lambda architecture allows for real-time processing of data, which is crucial in industrial environments where timely decision-making is critical.
3. Auto model selection: The auto model selection system used in the core module of the Network Traffic Analyzer ensures that the best-performing Machine Learning model is chosen, optimizing the system's performance for the specific use case.
4. Integration: The proposed architecture can be integrated with other cybersecurity tools, making it a comprehensive solution for detecting and mitigating cyber-attacks in industrial environments.

These features make it an optimal solution for the specific use case described in the paper.

5 Conclusions

This study presents a novel architectural standardization for the intelligent management and mitigation of sophisticated cyber threats. Specifically, it presents an innovative network traffic analyzer, a core module of the CTI2SA of the Cyber-pi project, that improves active industrial cybersecurity methods and significantly overcomes existing processes. CTI2SA is a fully interoperable and intelligent system that exploits cyber knowledge provided daily by cyber threat managers worldwide. This offers a high level of security for the supervised information infrastructure of an organization. Its architecture is built on standards that can sustain secure communication with dependable information sources and get regular updates on new and existing dangers. The upgrades above are initially prioritized and tailored to fit the information architecture of the supported organization. The operating systems of information systems are then aligned, and the automation rules are put into operation by implementing quick action alerts to the system administrators. It is an effective real-time technique for monitoring and promptly identifying occurrences, dramatically improving an organization's operational cyber security.

Based on the λ-Architecture, the proposed network traffic analyzer combines batch and stream processing to handle huge volumes of data while balancing latency, throughput, and fault tolerance. The core module of this analyzer employs a unique auto model selection approach that selects the highest performing ML model among competitors. The goal is to continually upgrade the vulnerability detection capabilities of the whole system. This cutting-edge research idea has never been proposed before in the literature, and we believe it has the potential to significantly advance the state of the art in machine learning-based industrial cybersecurity. Unfortunately, as far as we are aware, there is no comparable project against which we may compare this one. To minimize prejudice or false impressions, we describe the performance of the suggested model without comparing it to any other technique based on a comparative architectural framework. On the other hand, the practical contribution of this unique methodology relies on constructing an automated model selection system that chooses which machine learning algorithm to employ, including optimization of the suitable hyperparameters. As a result, an even more efficient, accurate, and updatable cyber defense procedure is produced in a simple and resilient manner without the need for cybersecurity and machine learning (ML) expert human supervision.

The proposed technique can be extended to cover a wider scientific area without reducing the main points currently described by adopting one or more of the following approaches:

1. Generalization: The proposed technique can be generalized to cover a wider scientific area by identifying the fundamental principles and concepts applicable across different domains. For instance, the concept of using batch and stream processing to manage large amounts of data can be applied to various fields, such as finance, healthcare, and e-commerce.
2. Adaptation: The proposed technique can cover a wider scientific area by modifying some specific features to suit the new domain's requirements. For example, the auto model selection system can be adapted to work with different machine learning algorithms commonly used in a particular field.
3. Integration: The proposed technique can be integrated with existing systems and tools to cover a wider scientific area. For instance, the Network Traffic Analyzer can be integrated with other cybersecurity tools, such as firewalls and intrusion detection systems, to provide a comprehensive solution for detecting and mitigating cyber-attacks.
4. Collaboration: The proposed technique can be extended to cover a wider scientific area by collaborating with experts from different domains. This approach can help identify the specific requirements and challenges in different fields and develop customized solutions that meet the specific needs of each domain.

The proposed approach to the Network Traffic Analyzer has certain limitations, including:

1. Limited evaluation: The proposed system's performance was evaluated using a single dataset, and there may be variations in performance when applied to other datasets. Therefore, further evaluation is necessary to determine the system's generalizability to other datasets.
2. Limited scope: The proposed system is tailored for the specific use case of detecting and mitigating cyber-attacks in industrial environments, and its applicability to other domains may be limited. Therefore, the proposed system's generalizability to other domains requires further investigation.
3. Computational resources: The proposed system requires significant computational resources to handle large amounts of data. The system's performance may be limited by the availability of computational resources, particularly in smaller organizations or those with limited IT infrastructure.
4. Cost: The proposed system may have a higher cost than other existing cybersecurity solutions, which may limit its adoption by some organizations, particularly smaller ones with limited budgets.
5. Real-world implementation: The proposed system has not been implemented in a real-world industrial environment, and its performance in such an environment may differ from that observed in the evaluation.

Finding ways to compare logs and security policies to hasten their convergence is initially the most important challenge for the development of the suggested system. It would also be a significant enhancement if CTI2SA were enhanced with more advanced anomaly detection algorithms that consider most of the organization's operational characteristics, such as job scheduling, local events, technical advancements or system adaptations. It is also important to look at the system's structure in the context of data transformation methods so that intelligent processes may determine the best ways to represent various kinds of structured and unstructured data to make it easier for self-healing rules to be applied. Last but not least, the CTI2SA's future growth must prioritize using interpretation models. These models may define individual predictions using methods like Shapley values and can explain the decision-making process, including the importance of characteristics and the accumulation of local effects. The goal of model interpretation is to translate the working processes of models into the human-understandable language to investigate adversarial assaults and defenses.

Acknowledgements Co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (project code: T2EDK-01469).

Author contributions Conceptualization was contributed by AP, AA, CI; methodology was contributed by AP, CI; software was contributed by AP, CI, KD, KR; validation was contributed by AP, AA, CI, KD, KR; formal analysis was contributed by AP, AA, CI, KR; investigation was contributed by AP, AA, CI; resources were contributed by AP, CI; data curation was contributed by AP, AA, CI, KD, KR; writing—original draft preparation, was contributed by AP, KD, KR; writing—review and editing, was contributed by AP, AA, CI, KD, KR; visualization was contributed by AP, AA, CI; supervision was contributed by CI, KR; project administration was contributed by AP; funding acquisition was contributed by AP, AA, CI. All authors have read, reviewed and agreed to the published version of the manuscript.

Data availability The data used in this study are available from the author upon request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. **17**(4), 2347–2376 (2015). https://doi.org/10.1109/COMST.2015.2444095
2. Harjula, E., Artemenko, A., Forsström, S.: Edge computing for industrial IoT: challenges and solutions. In: Mahmood, N.H., Marchenko, N., Gidlund, M., Popovski, P. (eds.) Wireless Networks and industrial IoT: applications, challenges and enablers, pp. 225–240. Springer International Publishing, Cham (2021)
3. Al Enany, M.O., Harb, H.M., Attiya, G.: A comparative analysis of MQTT and IoT application protocols. In: 2021 International Conference on Electronic Engineering (ICEEM), pp. 1–6 (2021)
4. Banafa, A.: 2 The Industrial Internet of Things (IIoT): challenges, requirements and benefits. In: Secure and Smart Internet of Things (IoT): Using Blockchain and AI, River Publishers, pp. 7–12. (2018). Accessed 19 Jan 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9226906
5. Boubekeur, M.: Industrial applications for cyber-physical systems. In: 2017 First International Conference on Embedded Distributed Systems (EDiS), pp. 59–59. (2017)
6. Chen, H., Hu, M., Yan, H., Yu, P.: Research on industrial internet of things security architecture and protection strategy. In: 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), pp. 365–368. (2019)
7. Geng, H.: The industrial internet of things (IIoT). In: Internet of Things and Data Analytics Handbook, Wiley, pp. 41–81. (2017)
8. Farooq, M.J., Zhu, Q.: IoT supply chain security: overview, challenges, and the road ahead. ArXiv190807828 Cs. (2019), Accessed 19 Jan 2021. [Online]. Available: http://arxiv.org/abs/1908.07828
9. Dawood, K.: An overview of renewable energy and challenges of integrating renewable energy in a smart grid system in Turkey. In: 2020 International Conference on Electrical Engineering (ICEE), pp. 1–6. (2020)
10. Khan, W.Z., Rehman, M.H., Zangoti, H.M., Afzal, M.K., Armi, N., Salah, K.: Industrial internet of things: recent advances, enabling technologies and open challenges. Comput. Electr. Eng. **81**, 106522 (2020). https://doi.org/10.1016/j.compeleceng.2019.106522
11. Rouhani, S., Deters, R.: Blockchain based access control systems: state of the art and challenges. IEEEWICACM Int. Conf. Web Intell. (2019). https://doi.org/10.1145/3350546.3352561
12. Choo, K.R., Gritzalis, S., Park, J.H.: Cryptographic solutions for industrial internet-of-things: research challenges and opportunities. IEEE Trans. Ind. Inform. **14**(8), 3567–3569 (2018). https://doi.org/10.1109/TII.2018.2841049
13. Mahalle, V.S., Shahade, A.K.: Enhancing the data security in Cloud by implementing hybrid (Rsa amp; Aes) encryption algorithm. In: 2014 International Conference on Power, Automation and Communication (INPAC), pp. 146–149. (2014)
14. Demertzis, K., Rantos, K., Drosatos, G.: A dynamic intelligent policies analysis mechanism for personal data processing in the IoT ecosystem. Big Data Cogn. Comput. **4**(2), 9 (2020). https://doi.org/10.3390/bdcc4020009
15. de Souza, P.V.C., Guimarães, A.J., Rezende, T.S., Souza Araujo, V., do Nascimento, L.A.F., Oliveira Batista, L.: An intelligent hybrid model for the construction of expert systems in malware detection. In: 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 1–8. (2020) doi: https://doi.org/10.1109/EAIS48028.2020.9122770.
16. Majed, H., Noura, H.N., Chehab, A.: Overview of digital forensics and anti-forensics techniques. In: 2020 8th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–5. (2020). doi: https://doi.org/10.1109/ISDFS49300.2020.9116399.
17. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.K.: A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues. IEEE Commun. Surv. Tutor. **22**(2), 1191–1221 (2020). https://doi.org/10.1109/COMST.2019.2962586
18. Rantos, K., Drosatos, G., Demertzis, K., Ilioudis, C., Papanikolaou, A., Kritsas, A.: ADvoCATE: a consent management platform for personal data processing in the IoT using blockchain technology. In: Lanet, J.-L., Toma, C. (eds.) Innovative Security Solutions for Information Technology and Communications Lecture Notes in Computer Science, pp. 300–313. Springer International Publishing, Cham (2019)
19. Choi, S., Yun, J.-H., Kim, S.-K.: A comparison of ICS datasets for security research based on attack paths. In: Luiijf, E., Žutautaitė, I., Hämmerli, B.M. (eds.) Critical Information Infrastructures Security Lecture Notes in Computer Science, pp. 154–166. Springer International Publishing, Cham (2019)
20. Rantos, K., Spyros, A., Papanikolaou, A., Kritsas, A., Ilioudis, C., Katos, V.: Interoperability challenges in the cybersecurity information sharing ecosystem. Computers **9**(1), 18 (2020). https://doi.org/10.3390/computers9010018
21. Rhoades, D.: Machine actionable indicators of compromise. In: 2014 International Carnahan Conference on Security Technology (ICCST), pp. 1–5. (2014) https://doi.org/10.1109/CCST.2014.6987016.
22. Akram, B., Ogi, D.: The making of indicator of compromise using malware reverse engineering techniques. In: 2020 International Conference on ICT for Smart Society (ICISS), pp. 1–6. (2020)
23. Atluri, V., Horne, J.: A machine learning based threat intelligence framework for industrial control system network traffic indicators of compromise. In: SoutheastCon 2021, pp. 1–5. (2021)
24. Verma, M., Kumarguru, P., Brata Deb, S., Gupta, A.: Analysing indicator of compromises for ransomware: leveraging IOCs with

machine learning techniques. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 154–159. (2018)

25. Garbis, J., Chapman, J.W.: Privileged access management. In: Garbis, J., Chapman, J.W. (eds.) Zero Trust Security: An Enterprise Guide, pp. 155–161. Apress, Berkeley (2021)
26. MicrosoftGuyJFlo, "Developing a privileged access strategy." https://learn.microsoft.com/en-us/security/compass/privileged-access-strategy (accessed 18 Sept 2022)
27. Moorhead Patrick, M.K.: RESEARCH PAPER: modern privileged access management. In: Moor Insights & Strategy, 26 Jan 2022. https://moorinsightsstrategy.com/research-paper-modern-privileged-access-management/ (accessed 18 Sep 2022)
28. Miller, D.J., Xiang, Z., Kesidis, G.: Adversarial learning targeting deep neural network classification: a comprehensive review of defenses against attacks. Proc. IEEE **108**(3), 402–433 (2020). https://doi.org/10.1109/JPROC.2020.2970615
29. Zhou, Z., Kuang, X., Sun, L., Zhong, L., Xu, C.: Endogenous security defense against deductive attack: when artificial intelligence meets active defense for online service. IEEE Commun. Mag. **58**(6), 58–64 (2020). https://doi.org/10.1109/MCOM.001.1900367
30. Xing, K., Li, A., Jiang, R., Jia, Y.: A review of APT attack detection methods and defense strategies. In: 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC), pp. 67–70. (2020)
31. Gupta, S.K., Tripathi, M., Grover, J.: Towards an effective intrusion detection system using machine learning techniques: comprehensive analysis and review. In: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1–6. (2021)
32. Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J.: Machine learning for networking: workflow, advances and opportunities. IEEE Netw. **32**(2), 92–99 (2018). https://doi.org/10.1109/MNET.2017.1700200
33. Llopis, S. et al.: A comparative analysis of visualisation techniques to achieve cyber situational awareness in the military. In: 2018 International Conference on Military Communications and Information Systems (ICMCIS), pp. 1–7. (2018)
34. Yang, B., Liu, D.: Research on network traffic identification based on machine learning and deep packet inspection. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1887–1891. (2019)
35. Trabelsi, Z., Zeidan, S., Masud, M.M.: Network packet filtering and deep packet inspection hybrid mechanism for IDS early packet matching. In: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), pp. 808–815. (2016)
36. Yang, Z., Sun, Q., Zhang, Y., Wang, W.: Identification of malicious injection attacks in dense rating and co-visitation behaviors. IEEE Trans. Inf. Forensics Secur. **16**, 537–552 (2021). https://doi.org/10.1109/TIFS.2020.3016827
37. Alhasan, S., Abdul-Salaam, G., Bayor, L., Oliver, K.: Intrusion detection system based on artificial immune system: a review. In: 2021 International Conference on Cyber Security and Internet of Things (ICSIoT), pp. 7–14. (2021)
38. Dhingra, M., Jain, M., Jadon, R.S.: Role of artificial intelligence in enterprise information security: a review. In: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 188–191. (2016)
39. Mohammed, A., George, G.: Vulnerabilities and strategies of cybersecurity in smart grid—evaluation and review. In: 2022 3rd International Conference on Smart Grid and Renewable Energy (SGRE), pp. 1–6. (2022)
40. Hota, A.R., Sundaram, S.: Interdependent security games on networks under behavioral probability weighting. IEEE Trans. Control Netw. Syst. **5**(1), 262–273 (2018). https://doi.org/10.1109/TCNS.2016.2600484
41. Goli, Y.D., Ambika, R.: Network traffic classification techniques-a review. In: 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), pp. 219–222. (2018)
42. Yu, M.J., Jung, J.H., Lee, J.S.: Design and implementation of a packet analyzer for traffic monitoring in tactical communication network. In: 2016 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1239–1241. (2016)
43. Addeen, H.H., Xiao, Y., Li, J., Guizani, M.: A survey of cyber-physical attacks and detection methods in smart water distribution systems. IEEE Access **9**, 99905–99921 (2021). https://doi.org/10.1109/ACCESS.2021.3095713
44. Kashinath, S.A., et al.: Review of data fusion methods for real-time and multi-sensor traffic flow analysis. IEEE Access **9**, 51258–51276 (2021). https://doi.org/10.1109/ACCESS.2021.3069770
45. Novakov, S., Lung, C.-H., Lambadaris, I., Seddigh, N.: Combining statistical and spectral analysis techniques in network traffic anomaly detection. In: 2012 Next Generation Networks and Services (NGNS), pp. 94–101. (2012)
46. Sinadskiy, A., Domukhovsky, N.: Statistical-entropy method for zero knowledge network traffic analysis algorithm implementation. In: 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), pp. 611–614. (2020)
47. Coulter, R., Han, Q.-L., Pan, L., Zhang, J., Xiang, Y.: Data-driven cyber security in perspective—intelligent traffic analysis. IEEE Trans. Cybern. **50**(7), 3081–3093 (2020). https://doi.org/10.1109/TCYB.2019.2940940
48. Thakare, S., Pund. A., Pund, M.A.: Network traffic analysis, importance, techniques: a review. In: 2018 3rd International Conference on Communication and Electronics Systems (ICCES), pp. 376–381. (2018)
49. Lazar, A., Wu, K., Sim, A.: Predicting network traffic using TCP anomalies. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 5369–5371. (2018)
50. Naing, M.T., Khaing, T.T., Maw, A.H.: Evaluation of TCP and UDP traffic over software-defined networking. In: 2019 International Conference on Advanced Information Technologies (ICAIT), pp. 7–12. (2019)
51. Hsu, C.-H., Huang, C.-Y., Chen, K.-T.: Fast-flux bot detection in real time. In: Jha, S., Sommer, R., Kreibich, C. (eds.) Recent Advances in Intrusion Detection Lecture Notes in Computer Science, pp. 464–483. Springer, Berlin (2010)
52. Rana, S., Aksoy, A.: Automated fast-flux detection using machine learning and genetic algorithms. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. (2021)
53. Haffner, P., Sen, S., Spatscheck, O., Wang, D.: ACAS: automated construction of application signatures. In: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, in MineNet '05. New York, NY, USA: Association for Computing Machinery, pp. 197–202. (2005)
54. Caglayan, A., Toothaker, M., Drapeau, D., Burke, D., Eaton, G.: Real-time detection of fast flux service networks. In: 2009 Cybersecurity Applications & Technology Conference for Homeland Security, pp. 285–292. (2009)
55. Ding, W., Ren, W., Xia, Z., Wang, L.: Botnet tracing based on distributed denial of service activity analysis. In: 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI), pp. 685–689. (2015)
56. Tsai, M.-H., Chang, K.-C., Lin, C.-C., Mao, C.-H., Lee, H.-M.: C&C tracer: Botnet command and control behavior tracing.

In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1859–1864. (2011)

57. Wang, Z., Fok, K.-W., Thing, V.L.L.: Machine learning for encrypted malicious traffic detection: approaches, datasets and comparative study. Comput. Secur. **113**, 102542 (2022). https://doi.org/10.1016/j.cose.2021.102542
58. Jorgensen, S. et al.: Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification. arXiv, May 11, (2022) doi: https://doi.org/10.48550/arXiv.2205.05628
59. Chaabane, A., Manils, P., Kaafar, M.A.: Digging into anonymous traffic: a deep analysis of the tor anonymizing network. In: 2010 Fourth International Conference on Network and System Security, pp. 167–174. (2010)
60. Ishitaki, T., Obukata, R., Oda, T., Barolli, L.: Application of deep recurrent neural networks for prediction of user behavior in tor networks. In: 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 238–243. (2017)
61. Juan, W., Shimin, C., Jun, Z., Bin, H., Lei, S.: Identification of tor anonymous network traffic based on machine learning. In: 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 150–153. (2021)
62. Gao, Y., Li, X., Peng, H., Fang, B., Yu, P.: HinCTI: a cyber threat intelligence modeling and identification system based on heterogeneous information network. IEEE Trans. Knowl. Data Eng. (2020). https://doi.org/10.1109/TKDE.2020.2987019
63. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in KDD '17. Association for Computing Machinery, pp. 635–644. (2017)
64. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, in CCS '16. Association for Computing Machinery, pp. 755–766. (2016)
65. Modi, A. et al.: Towards automated threat intelligence fusion. In: 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC), pp. 408–416. (2016)
66. Gascon, H., Grobauer, B., Schreck, T., Rist, L., Arp, D., Rieck, K.: Mining attributed graphs for threat intelligence. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, in CODASPY '17. Association for Computing Machinery, pp. 15–22. (2017)
67. Sengupta, S., Chowdhary, A., Huang, D., Kambhampati, S.: General sum markov games for strategic detection of advanced persistent threats using moving target defense in cloud networks. In: Alpcan, T., Vorobeychik, Y., Baras, J.S., Dán, G. (eds.) Decision and Game Theory for Security Lecture Notes in Computer Science, pp. 492–512. Springer International Publishing (2019)
68. Bhatt, S., Manadhata, P.K., Zomlot, L.: The operational role of security information and event management systems. IEEE Secur. Priv. **12**(5), 35–41 (2014). https://doi.org/10.1109/MSP.2014.103
69. Introduction to STIX. https://oasis-open.github.io/cti-documentation/stix/intro.html (accessed 14 Oct 2021)
70. Spyros, A., Rantos, K., Papanikolaou, A., Ilioudis, C.: An innovative self-healing approach with STIX data utilization. In: Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, pp. 645–651. SCITEPRESS - Science and Technology Publications, Lieusaint - Paris, France, (2020)
71. Guillen, E., Padilla, D., Colorado, Y.: Weaknesses and strengths analysis over network-based intrusion detection and prevention systems. In: 2009 IEEE Latin-American Conference on Communications, pp. 1–5. (2009)
72. Özer, E., İskefiyeli, M.: Detection of DDoS attack via deep packet analysis in real time systems. In: 2017 International Conference on Computer Science and Engineering (UBMK), pp. 1137–1140. (2017)
73. OSSEC - World's Most Widely Used Host Intrusion Detection System—HIDS. OSSEC. https://www.ossec.net/ (accessed 14 Oct 2021)
74. MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing (formely known as Malware Information Sharing Platform)." https://www.misp-project.org/ (accessed 14 Oct 2021)
75. Yang, Y. et al.: Dark web forum correlation analysis research. In 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 1216–1220. (2019)
76. Spagnolo, G.S., Santis, M.D.: Computer generated hologram for SemiFragile watermarking with encrypted images. Int. J. Comput. Inf. Eng. **2**(11), 3829–3837 (2008)
77. Ye, G., Jiao, K., Wu, H., Pan, C., Huang, X.: An asymmetric image encryption algorithm based on a fractional-order chaotic system and the RSA public-key cryptosystem. Int. J. Bifurc. Chaos (2020). https://doi.org/10.1142/S0218127420502338
78. Habibi Lashkari, A., Kaur, G., Rahali, A.: DIDarknet: a contemporary approach to detect and characterize the darknet traffic using deep image learning. In: 2020 the 10th International Conference on Communication and Network Security, in ICCNS 2020. pp. 1–13. Association for Computing Machinery, New York, NY, USA (2020)
79. Fan, J., Mu, D., Liu, Y.: Research on network traffic prediction model based on neural network. In: 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), pp. 554–557. (2019)
80. Alghamdi, R., Bellaiche, M.: A deep intrusion detection system in lambda architecture based on edge cloud computing for IoT. In: 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), pp. 561–566. (2021)
81. Sanla, A., Numnonda, T.: A comparative performance of real-time big data analytic architectures. In: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 1–5. (2019)
82. Zahid, H., Mahmood, T., Morshed, A., Sellis, T.: Big data analytics in telecommunications: literature review and architecture recommendations. IEEECAA J. Autom. Sin. **7**(1), 18–38 (2020). https://doi.org/10.1109/JAS.2019.1911795
83. Suthakar, U., Magnoni, L., Smith, D.R., Khan, A.: Optimised lambda architecture for monitoring scientific infrastructure. IEEE Trans. Parallel Distrib. Syst. **32**(6), 1395–1408 (2021). https://doi.org/10.1109/TPDS.2017.2772241
84. Hoseiny Farahabady, M., Taheri, J., Tari, Z., Zomaya, A.Y.: A dynamic resource controller for a lambda architecture. In: 2017 46th International Conference on Parallel Processing (ICPP), pp. 332–341. (2017)
85. Batyuk, A., Voityshyn, V., Verhun, V.: Software architecture design of the real- time processes monitoring platform. In: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), pp. 98–101. (2018)
86. Parres-Peredo, A., Piza-Davila, I., Cervantes, F.: Building and evaluating user network profiles for cybersecurity using serverless architecture. In: 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), pp. 164–167. (2019)
87. Ge, P.: Analysis on approaches and structures of automated machine learning frameworks. In: 2020 International Conference

on Communications, Information System and Computer Engineering (CISCE), pp. 474–477. (2020)

88. Nagarajah, T., Poravi, G.: A review on automated machine learning (AutoML) systems. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), pp. 1–6. (2019)
89. Nguyen, D.A., Kononova, A.V., Menzel, S., Sendhoff, B., Bäck, T.: An efficient contesting procedure for AutoML optimization. IEEE Access **10**, 75754–75771 (2022). https://doi.org/10.1109/ACCESS.2022.3192036
90. Nguyen, D.A., Kononova, A.V., Menzel, S., Sendhoff, B., Back, T.: Efficient AutoML via combinational sampling. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01–10. (2021)
91. Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K.: Auto-WEKA: automatic model selection and hyperparameter optimization in WEKA. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated Machine Learning: Methods, Systems, Challenges The Springer Series on Challenges in Machine Learning, pp. 81–95. Springer International Publishing, Cham (2019)
92. Feurer, M., Hutter, F.: Hyperparameter optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated Machine Learning: Methods, Systems, Challenges The Springer Series on Challenges in Machine Learning, pp. 3–33. Springer International Publishing, Cham (2019)
93. Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., Aguilar, J.: Towards the deployment of machine learning solutions in network traffic classification: a systematic survey. IEEE Commun. Surv. Tutor. **21**(2), 1988–2014 (2019). https://doi.org/10.1109/COMST.2018.2883147
94. Dong, H., Munir, A., Tout, H., Ganjali, Y.: Next-generation data center network enabled by machine learning: review, challenges, and opportunities. IEEE Access **9**, 136459–136475 (2021). https://doi.org/10.1109/ACCESS.2021.3117763
95. Jirsik, T.: Stream4Flow: Real-time IP flow host monitoring using Apache Spark. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–2. (2018)
96. Jirsik, T., Celeda, P.: Toward real-time network-wide cyber situational awareness. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–7. (2018)
97. Shi, J., Leau, Y.-B., Li, K., Park, Y.-J., Yan, Z.: Optimization and decomposition methods in network traffic prediction model: a review and discussion. IEEE Access **8**, 202858–202871 (2020). https://doi.org/10.1109/ACCESS.2020.3036421
98. Saxena, A., Pant, B., Alanya-Beltran, J., Akram, S.V., Bhaskar, B., Bansal, R.: A detailed review of implementation of deep learning approaches for industrial internet of things with the different opportunities and challenges. In: 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), pp. 1370–1375. (2022)
99. Macas, M., Wu, C.: Review: deep learning methods for cybersecurity and intrusion detection systems. In: 2020 IEEE Latin-American Conference on Communications (LATINCOM), pp. 1–6. (2020)
100. Halbouni, A., Gunawan, T.S., Habaebi, M.H., Halbouni, M., Kartiwi, M., Ahmad, R.: Machine learning and deep learning approaches for cybersecurity: a review. IEEE Access **10**, 19572–19585 (2022). https://doi.org/10.1109/ACCESS.2022.3151248
101. Das, A., Balakrishnan, S.G.: A comparative analysis of deep learning approaches in intrusion detection system. In: 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pp. 555–562. (2021)
102. Hamouda, D., Ferrag, M.A., Benhamida, N., Seridi, H.: Intrusion detection systems for industrial internet of things: a survey. In: 2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS), pp. 1–8. (2021)
103. Dicholkar, S.V., Sekhar, D.: Review-IoT security research opportunities. In: 2020 International Conference on Convergence to Digital World - Quo Vadis (ICCDW), pp. 1–4. (2020)
104. Dmitrievich, A.G., Nikolaevich, S.A.: Automated process control anomaly detection using machine learning methods. In: 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), pp. 0536–0538. (2020)
105. Ali, R.F., Muneer, A., Dominic, P.D.D., Ghaleb, E.A.A., Al-Ashmori, A.: Survey on cyber security for industrial control systems. In: 2021 International Conference on Data Analytics for Business and Industry (ICDABI), pp. 630–634. (2021)
106. Tsiknas, K., Taketzis, D., Demertzis, K., Skianis, C.: Cyber threats to industrial IoT: a survey on attacks and countermeasures. IoT **2**, 1 (2021). https://doi.org/10.3390/iot2010009

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.