Journal of Cloud Computing:
Advances, Systems and Applications

REVIEW

Open Access

Complex event processing for physical and cyber security in datacentres - recent progress, challenges and recommendations

Khaled A. Alaghbari[1,2], Mohamad Hanif Md Saad[2,3*], Aini Hussain[4] and Muhammad Raisul Alam[2,5]

Abstract

A datacentre stores information and manages data access in fast and reliable manner. Failure of datacentre operation is not an option and can be catastrophic. Internet of things (IoT) devices in datacentre can automate management tasks and reduce human intervention and error. IoT devices can be used to manage many datacentre routine tasks such as monitoring physical infrastructure, updating software and configuration, monitoring network traffic, and automating alerting reports to respective authorities. The physical and cyber security of the datacentre can be handled by IoT technology by intrusion detection methods. By 2025, more than 25 billion things will be connected to the internet network, therefore massive data will be generated by different heterogeneous sources, and powerful processing engines such as complex event processing (CEP) are needed to handle such a fast and continuous stream of big data. The integration of machine learning (ML) and deep learning (DL) can enhance CEP by introducing new features such as automated rule extraction and self-healing mechanism. This study aims to provide an overview of CEP, as well as its features and potential for integration with IoT applications and ML/DL techniques. We provide a review of recent research works to highlight the capability and applicability of CEP technology to monitor physical facilities and cyber security in detail. This review also highlights several issues and challenges, and provides suggestions for future research. The highlighted insights and recommendations in this paper could raise efforts toward the development of future datacentres based on CEP technology.

Keywords: Datacentre, Physical security, Cyber security, IoT, CEP, Machine learning, Deep learning, Intelligent systems

Introduction

In the era of the fourth industrial revolution (IR4.0), computer network systems (CNS) form a critical infrastructure in the connected world. It allows the exchange of information between remote sites and facilitates communication between organizations such as government agencies, universities, medical facilities, military bases, social networking services such as Google and Facebook, and many other organisations that rely on CNS to function properly. CNS facilities such as datacentre servers and communication gateway are very critical to ensure that they are in good condition. The security of datacentres is critical for organizations; there should be a minimum or no chance of a security breach. The datacentres must maintain equal attention to both facility and the hosted equipment in order to meet their objectives. Failure of the datacentre servers is not an option and can have disastrous consequences, for instance failure of physical security system to warn about high temperature can cause damage to hardware, low or high humidity can lead to issues like corrosion of metal components, static electrical discharge, and water damage to equipment, breach of cyber security can cause stealing customers personal information, financial losses, reputational

*Correspondence: hanifsaad@ukm.edu.my

[3] Department of Mechanical & Manufacturing Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia
Full list of author information is available at the end of the article

damage to the organization and even paying penalties for failing to comply with data protection legislation [1–3].

The Internet of things (IoT) is a promising technology to drive IR4.0 since it has the ability to transfer information over a network without the need for human-to-human or human-to-computer interaction. The goal of IoT is to provide smart environments to improve the efficiency of monitoring and detection [4, 5]. Datacentre facility-related sensors such as temperature, humidity, door, voltage supply, smoke, aircon state, and fan state sensors from various sources need to be acquired in real-time and processed to identify related events (e.g., the server door opened and closed in time, high temperature in network gateway disrupting transmission process, power disruption in server rooms disturbing the network, backup batteries are not charged properly). Moreover, network traffic and status should be monitored to prevent unusual behaviours and hacks. Through a network attack, hackers may attempt to leak sensitive data in individual cloud networks or a local area network. Intrusion detection system (IDS) is a critical tool for cyber security specialists to deal with attacks and protect networks and secret data. Anomaly detection can assist in detecting network intrusion attacks in real-time. Then smart datacentre automatically notifies the technical support team for detecting anomalous physical or network events via SMS (short message service) or other available source and instructs them to take mitigation action. A more intelligent approach is to apply a self-healing mechanism for IoT devices to provide appropriate remediation without human intervention [6].

The IoT devices are designed to perform advanced services such as smart traffic, monitoring and control of the environment, and high definition (HD) video streaming. The number of IoT devices is expected to reach 25 billion by 2025 [7]. However, all the advantages of connectivity come with a certain amount of challenge and risk. The big data collected from the massive number of IoT devices rise great challenges for data processing. Complex event processing (CEP) is an emerged technology that analyses, filters, and matches semantically low-level data in order to identify complex events. The concept is first introduced by Luckman [8, 9] and then further elaborated by many researchers. CEP systems are based on the idea of identifying events in real-time by studying cause/effect links among simple events that contain no special information in unrelated conditions. The IoT is one of the primary domains that successfully employ the monitoring features of CEP [10]. CEP is now widely used in a variety of fields, including environmental monitoring [11, 12], industrial production monitoring [13], telemedicine monitoring [14], and building safety monitoring [15, 16]. There are limited studies that applied IoT for datacentres to provide physical and cyber security. In this paper, we review recent techniques that applied for near-field application and could be used for monitoring future datacentres. We focus on CEP technology and its possible integration with machine learning (ML) and deep learning (DL).

The main objective of this review paper is to attract the attention of the researchers to the advantages of CEP technology such as the ability to process massive real-time information, quick time-response when a complex event occurs and capability of CEP technology to be integrated with ML/DL techniques. The datacentre is used as a case study to identify the potential of using CEP with ML/DL for physical and cyber security, and provide recommendations to address these challenges. We have conducted a traditional literature review to identify and analyse the current and state-of-art research, the reviewed papers are collected from Google search engine based on different databases such as Scopus, Web of Science and IEEE. The keywords used for searching were: CEP + physical/cyber security, CEP + machine learning, CEP + deep learning, CEP + ML/DL + physical/cyber security, CEP + ML + security, IDS + CEP and IDS + CEP + ML in original names or abbreviations. A roadmap of our conducted review paper is illustrated in Fig. 1.

In the literature, few surveys were conducted for IoT security issues based on CEP, where most of the studies focus on IoT security issues based on ML and DL [4, 17–22]. It makes up for urgent needs to review current security risks and discuss the challenges and possible solutions based on CEP technology. Table 1 shows the research gap in the previous research and highlights our contributions. It shows the comparison between the previous survey papers and our survey with respect to CEP involvement in IoT technology, security issues related to physical or cyber attacks, and intrusion detection based on CEP, ML, DL, or a combination of CEP and ML. This survey paper presents a discussion on physical and cyber threats that could encounter datacentres to various intrusion techniques with a focus on CEP applications. The major contributions of our research are that it includes a detailed review of the recent progress in intrusion detection systems based on CEP and the current advancement of ML in the CEP domain.

The rest of the paper is organized as follows: Overview of datacentre, IoT and CEP section presents an overview of threats to the datacentres, IoT architecture and types of cyber-attacks. The CEP principles, event processing, available CEP engines in the market, applications, and the potential of IoT and CEP integration are presented at end of this section. Datacentre security section highlights the important aspects needed for designing a robust security

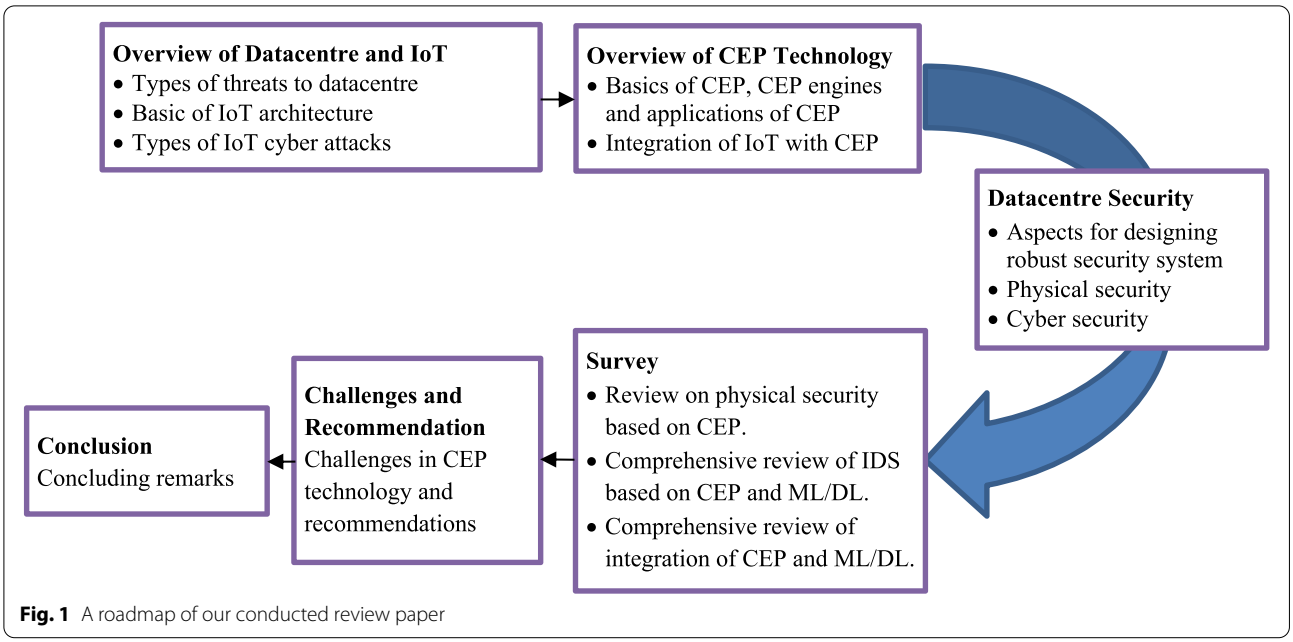


Fig. 1 A roadmap of our conducted review paper

system, followed by a comprehensive review of intrusion detection for physical and cyber security based on CEP and other techniques such as ML and DL. The potential of integrating CEP with ML is them reviewed. The challenges in CEP deployment and recommendations for future works are given in Challenges and recommendations section. Finally, Conclusion section concludes the paper.

Overview of datacentre, IoT and CEP

This section highlights the physical architectures of the datacentre, the architecture of IoT, and components of CEP. Then integration advantages of CEP with IoT are discussed.

Datacentre and security threats

A datacentre is a physical structure that is used to house critical data and applications. The switches, routers, firewalls, servers, storage systems, and application-delivery controllers are all essential components of a data centre design. Because these components are used to store and manage critical information and applications, datacentre security is a vital part of the design process [23]. The server room is typically physically isolated from the rest of the building, normally has no windows, and no one enters it except for server maintenance.

Threats to datacentres can be classified into two broad categories based on whether they are in the realm of IT software and networking (**Cyber threats**) or the realm of the data center's physical support infrastructure (**physical threats**) [19]. Hackers, viruses, network bottlenecks, and other accidental or malicious attacks on data security or flow are examples of cyber threats. Cyber threats have a prominent profile in the industry and the press, and most datacentres have different maintained systems to defend against them, such as firewalls and virus checkers. Power and cooling problems, human fault, leaks, fire, air quality, unauthorized personal access, not closing the server door properly are all examples of physical threats to the datacentre servers [24]. Table 2 shows examples of physical threats to the datacentre servers and the types of sensor that are needed to detect the threats.

IoT architecture and security threats

The basic IoT architecture consists of three layers which were introduced in the early stage of the research in this area [25]. The three fundamental layers are perception, network, and application layers [26, 27]. It has been then extended to five layers which include processing and business layers [28–30] as shown in Fig. 2. A number of IoT architectures have been suggested by researchers in the literature such as six and seven layers [31, 32] to address the IoT needs such as interoperability, scalability, security, privacy, and quality of service (QoS). Here we discussed the most important five layers.

i) **Physical/Perception layer**: This layer contains the sensors, actuators, and edge devices that are used to sense, collect information and interact with the desired environment.
ii) N**etwork/Transport/Connectivity layer**: This layer is responsible for transferring the collected sensor data

Table 1 A comparison of state-of-the-art surveys in IoT technology for intrusion detection

Paper	Year	Contribution	Technol.		Intrusion Detection					
			CEP	IoT	Type		Technique			
					Cyber	Physical	CEP	ML	DL	CEP + ML
[17] Tawsif et al.	2018	Review CEP solutions in the market from 2010 to 2017. Authors found majority of available CEP solutions are query-based and only a few work with ML.	✓	✓	✘	✘	✘	✘	✘	✘
[18] Wanner et al.	2019	Synthesis of research on the integration of CEP and ML identifying supervised learning as the predominant approach, and transfer of potentials for the use in smart factories.	✓	✓	✘	✘	✓	✓	✘	✓
[16] Hanif et al.	2020	Highlight challenges and issues of conventional event processing systems used in smart building, and present recommendations for advanced CEP systems.	✓	✓	✘	✘	✘	✘	✘	✘
[19] Al-Garadi et al.	2020	Provide surveys on ML techniques and recent advances in DL techniques that can be used for enhancing security techniques for IoT applications.	✘	✓	✓	✓	✘	✓	✓	✘
[20] Mishra et al.	2021	Discussion on cyber security issues in IoT such as DDoS, their impact and solutions based on anomaly detection using ML and DL.	✘	✓	✓	✘	✘	✓	✓	✘
Our paper	2022	Overview of physical and cyber threats to the datacentre, the potential of integration IoT with CEP, or CEP+ML. A comprehensive review of Intrusion detection based on CEP.	✓	✓	✓	✓	✓	✓	✓	✓

Table 2 Summary of the physical threat to the datacentre server room

Threat	Definition	Impact on datacentre	Types of sensors
Air temperature	The temperature of the air in the datacentre room, rack, and equipment.	Temperatures above specification and/or drastic temperature changes cause equipment failure and reduce equipment life span.	Temperature sensors
Humidity	At a specific temperature, the relative humidity of the datacentre room and rack.	- Static electricity buildup causes equipment failure at low humidity levels. - Formation of condensation at high humidity levels.	Humidity sensors
Leaks of liquid	Water or coolant leaks	Air conditioner problems are indicated by liquid damage to floors, cabling, and equipment.	- Spot leak sensors - Rope leak sensors
Smoke/Fire	Electrical or material fire	- Failure of equipment - Asset and data loss	- Supplement smoke sensors
Human error and unauthorised access	- Personnel's accidental wrongdoing - Unauthorized and/or coerced access to the datacentre with malicious intent	- Data loss and equipment damage - Downtime of equipment - Equipment theft and sabotage	- Motion sensors - Digital video cameras - Door contacts - Vibration sensors - Glass-break sensors
Dangerous airborne contaminants	Chemicals in the air, i.e., hydrogen from batteries, and particles such as dust	- Hazardous situation for workers and UPS (uninterruptible power supply), unreliability, and failure due to hydrogen release. - Failure of equipment due to increased static electricity and filter/fan clogging due to dust buildup	- Chemical/hydrogen sensors - Dust sensor

to the processing layer and vice versa through networks such as local area networks (LAN) and wide area networks (WAN) using wired or wireless technologies such as WiFi (wireless fidelity), Ethernet, Bluetooth, near field communications (NFC), ZigBee, cellular networks and low-power wide-area network (LPWAN).

iii) **Middleware/Processing layer**: Data accumulation and abstraction are the two main stages in this layer. The layer is used to capture, store, analyse and process massive amounts of data coming from the network layer. Moreover, it can be used to manage and provide different services to the other lower layers. Many techniques such as database, cloud computing, and big data processing modules can be employed in this layer. The middleware layer should enable cooperation between heterogeneous IoT devices and provide interoperability and scalability. In addition, the function of this layer is to provide security and privacy to IoT devices. A middleware layer should construct a mechanism to provide such tasks [19, 33]. CEP is the key part of the middleware layer which can help to obtain semantic meanings from primitive events based on certain rules [34]. ML/DL techniques can be integrated with CEP to simplify the event processing, the automatic generation of rule patterns and producing logical response as it will be discussed later.

iv) **Application layer**: This layer is responsible for delivering specific services demanded by the users, such as temperature, humidity light intensity, air pressure measurements. The application layer also offers data services such as big data storage, data warehousing, data mining, etc. for sematic data analysis. Smart building, smart healthcare, intelligent transportation system, and smart cities are examples of applications with smart user interfaces at the application layer.

v) **Business layer**: This layer manages the whole IoT system by providing business models, graphs, and flowcharts for the processed data received from the application layer. In addition, this layer supports automatic decision-making and the development of intelligent business strategies, based on big data analysis.

Several security threats and attacks may occur in each layer of IoT systems. Many attacks and crime targets the confidentiality of perception layer such as malicious fake code data injection, node capturing, and side-channel attack [33, 35]. The most important cyber security issues in network layer are confidentiality, compatibility and privacy. Phishing, denial-of-service (DoS), distributed DoS (DDoS), and data routing attacks (i.e., wormhole attack and sinkhole attack) are examples of attacks targeting this layer. Middleware layer is also vulnerable to several attacks such as cloud

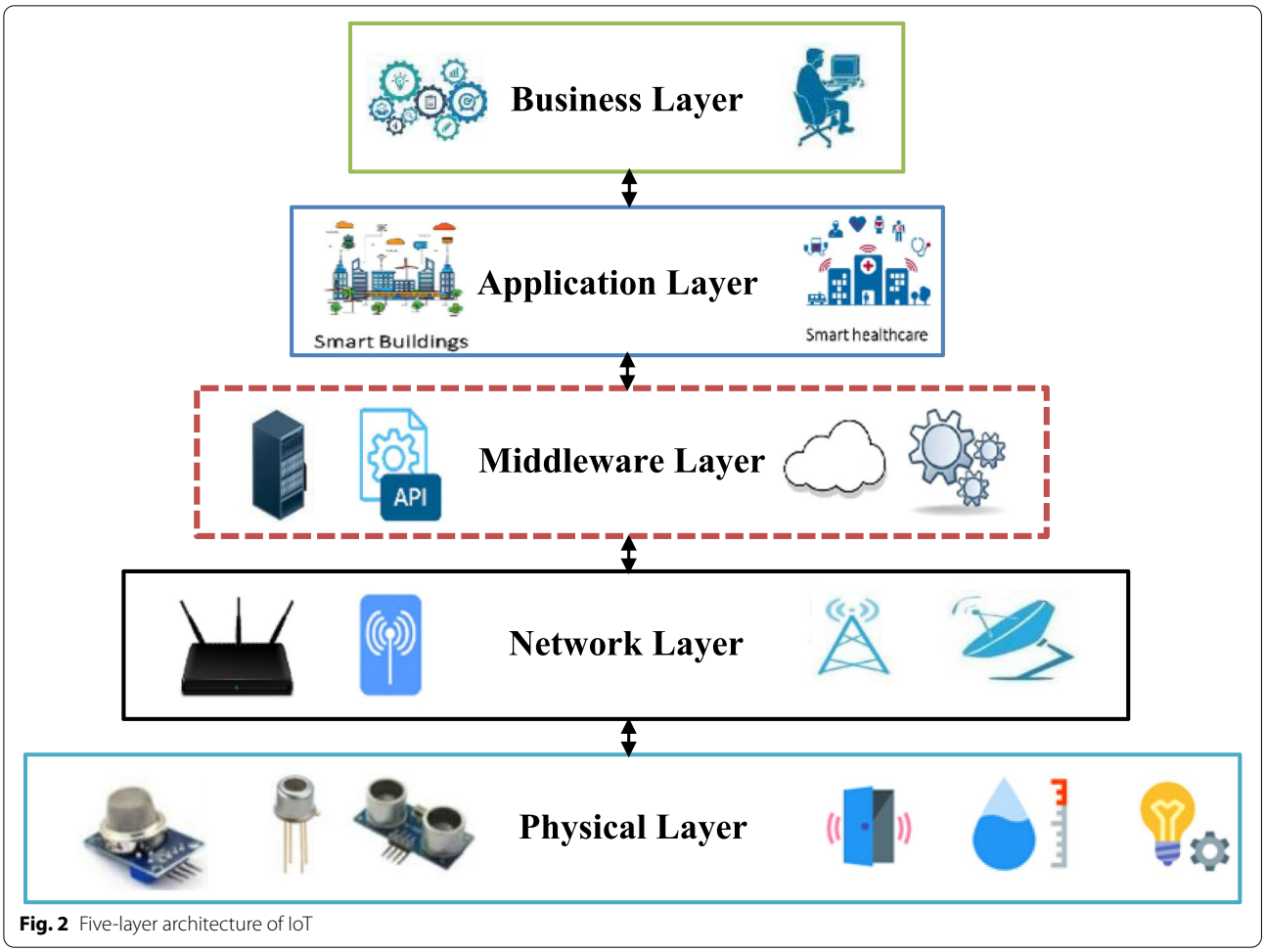


Fig. 2 Five-layer architecture of IoT

flooding attack, cloud malware injection, SQL injection attack, and man-in-the-middle attacks [35]. Access control attack, malicious code attack, sniffer attack, software bugs and service disruption are example of security challenges that need to be handled at the application layer. Through the use of viruses, worms, Trojan horses, and spyware, the attacker injects malware into the system in order to prevent service, manipulate data, or gain access to private information [33]. Table 3 describes several attacks that could be occurred

Table 3 Summary of different attacks in IoT system [36]

Attack Name	Description
DoS attack	Illegitimate users prevent legitimate users from accessing any device or service by flooding that device or service with false incoming requests.
Sniffer attack	Packet or sensitive information is captured by hackers using an application during the transmission stage in network, if it isn't adequately encrypted.
IP spoofing	Source IP addresses are added to data packet headers to hijack a network-connected user's browser.
Malware attack	A piece of software that performs malicious activities on the machines of users without their knowledge. Spyware and ransomware are two examples.
Wormhole attack	Malicious nodes in an IoT network build a tunnel, deceiving other nodes into thinking the two malicious nodes are close to each other and thus attracting data packets. This causes routing issues, and malicious nodes can tamper with data packets.
Sinkhole attack	During data transmission, a malicious node attacks a neighbouring nodes in order to reduce their efficiency.
Black hole attack	The malicious node receives data packets from neighbouring nodes but discards them.
Man-in-middle attack	Unknown person or malicious intrudes between the sender and receiver's conversation and attempts to access confidential information.

on IoT system. In today's world, botnets are responsible for the majority of massive cyber-attacks, such as DDoS attacks as well as spamming, identity theft, and other forms of fraud.

The name Botnet is derived from robot and network words. The Botnets are a collection of compromised devices known as Bots that are remotely controlled by a Botmaster using command and control (C&C) infrastructure as shown in Fig. 3 or even using a peer-to-peer (P2P) network. The Bots can be any collection of susceptible hosts such as computers, mobile phones, tablets, or IoT devices whose security have been breached. The C&C channels are used by the Botmaster to monitor and control his Bots to carry out malicious activities on specific targets. Hence the C&C channels play a key role in the communication within the network of the Botmaster and Bots [37]. Mirai is an example of botnets that has lately caused large-scale DDoS attacks by exploiting IoT devices due to its open source code which allows an attacker to create new variations [19]. Honeynets, signature-based detection, and anomaly detection models are some of the current Botnet analysis and detection methods [38]. Honeynets are used as traps to gather information about Bots and analyse their behaviour [39] when the mechanism of the monitored Bots has been revealed, a dedicated detection and blocking mechanism can be designed. Signature-based methods depend on a previously learned signature database of notorious Botnets. More discussion will be provided in the Cyber security section.

Complex event processing (CEP)

A CEP engine is a software platform that consists of a set of methods and representations of knowledge [40]. IoT monitoring needs a CEP engine to monitor the status of the system including event arrival. A large number of continuous data at low latency must be processed by CEP engines and should be able to quickly restore when the system fails. CEP deals with events, it is important to know the events types and how they are handled by CEP. In this section we discuss the type of events and their processing stages involved in CEP engine, then popular CEP engines are discussed and several potential applications are highlighted.

Type of events

The acquisition of IoT for events can be classified as **raw events**, **simple events (SEs)**, and **complex events (CEs)**. A raw event is an atomic (primitive) event data that is detected by a sensor device at a certain time. In a short period of time, IoT monitoring system can produce a high number of repeated, fragmented and redundant raw events. For instance, an RFID tag may be read multiple times throughout a very short period of time, however the application is only interested in whether the RFID tag is used during a certain period of time. As a result, raw data should be filtered and aggregated in real-time to produce significant SEs that reflect the status of one device or a collection of devices.

A simple event (SE) is typically mined from the raw events by filtering irrelevant events such as spatial repeated events generated by one sensing device at a very short time. The raw events that are not interested

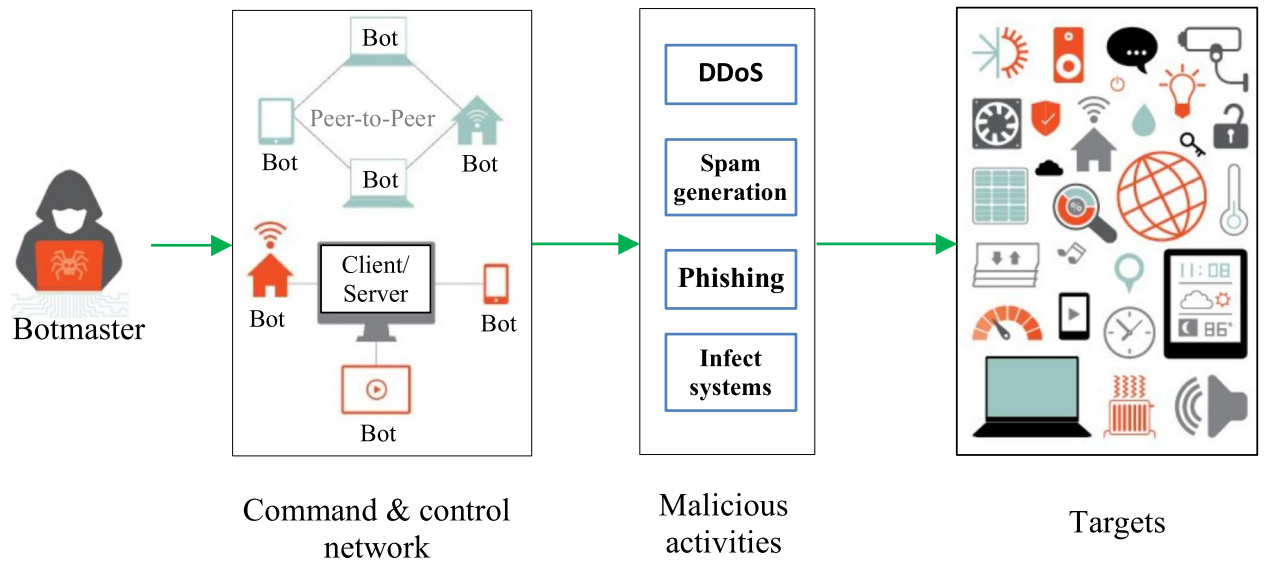


Fig. 3 The flow of Botnet attack from the Botmaster to the target devices across the command & control network

by applications can be neglected, for example the application may focus on certain data events at a specific moment in a specific area. Simple events merely represent simple facts, the application system, on the other hand, is more interested with the spatial and temporal correlation between groups of SEs, referred to as a CE. Complex (composite) events are formed by combining several SEs [41] after an analysis of the connection between SEs, such as time relationship, membership, inclusion relationship, causation, etc., and data processing, such as filtering, combining, and so on. CEP engine is the primary unit for determining whether SEs comply with a business rule. The IoT perceived event process from event production to event consumption is depicted in Fig. 4. First, sensor devices generate raw events. SEs are then created by filtering and composing raw events. Generally raw events are transformed into SEs that share a format like JSON, CSV, XML, etc. This is required by most of CEP engines before processing them. SEs are used to create CEs by matching event patterns. The CEs are delivered to applications to consume them. Between sensing devices and applications is where complex event processing (CEP) takes place. Small-grained events can be used to infer both simple and CEs. The difference is that SE inference rules are universal, SE granularity is moderate, and SE reusability is high, but CE inference rules are customised and defined by particular business requirements [15].

Event processing

Event processing is a technique of tracking and analysing (processing) streams of information about what happened (events) and drawing conclusions from them. As shown in Fig. 5, the first step in event processing is to **capture events** from source systems. Once a proper architecture is built, any system can be used as an event source. Web sites, credit card systems, phone networks, and call centres are just a few examples of real-world event sources. The following stage, **parsing and filtering**, is in charge of extracting business events from raw data. Raw events are filtered, parsed, and formatted in this step before being sent to the processing engine. Although an event carries crucial information, we also have critical information that is stored in other systems and not carried with the event. During the **Enrichment** phase, business events are enriched with historical data collected from external sources such as relational databases, NoSQL systems or web services. The system's heart is the **event processing** step. It processes streaming events and decides to act only when all of the necessary requirements are met. The event processing engine operates in accordance with predefined scenarios. The scenarios can be thought of as rule sets designed to reflect business requirements. Because every event is processed in memory, the events can be processed in real time. All these steps, capturing, parsing, enriching and processing events, are done for taking actions. Actions are connection points of event processing with other systems or users, for example sending SMS/email, calling user, sending notifications to App., or producing alarm. All these key steps should be **monitored** for operational and reporting purposes through a centralised system.

Popular CEP engines

There are many CEP engines available in the market as commercial or open source such as Esper/NEsper,

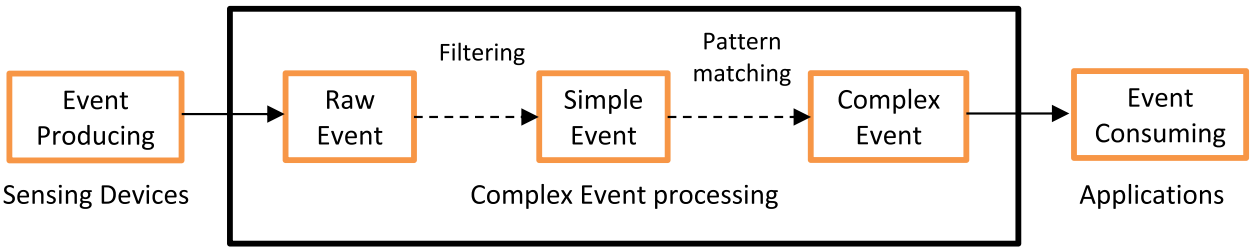


Fig. 4 IoT perceived event processing method

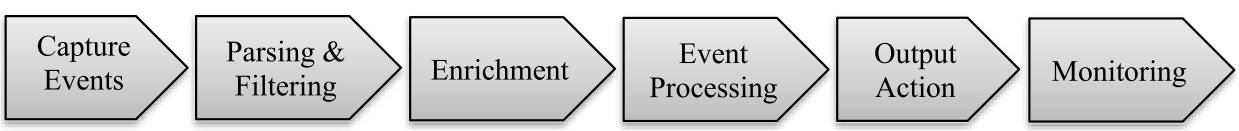


Fig. 5 Event processing stages

Drools, Progress Apama, Rulecore, Corel8 and CAISER [17]. Table 4 shows a number of modern CEP engines. Esper is an open source for CEP and event stream processing available for Java as Esper and for .NET as NEsper. Esper employs rule-based as an inference engine using a simple SQL-like language called event processing language (EPL). It uses application programming interface (API) such as API C# and API Java in the platform development [42]. Drools is an open source system written in Java, that provides a unified and integrated platform for rules, workflow and event processing. Drools engine is a rule-based inference engine where rule files are represented by Drools rule Language (DRL). Siddhi is a CEP engine that has been integrated with WSO2, and provided as open source. It is written in Java and thoroughly optimized for high performance, Siddhi query language (SiddhiQL) is language supported by Siddhi CEP engine [43]. It is important to mention that the EPL for each engine is different, for instance Esper EPL is the language supported by Esper, and SiddhiQL is the one supported by the Siddhi CEP engine. Most of these CEP engines use sliding window technique to detect complex events from SE sequence. CAISER (computer assisted intelligent event processor) [44] is a CEP platform that is suitable for developing CEP systems for engineering intelligent applications for integrating automatic event detection (sensor values, device status), processing SE and CE, event management and executing fast mitigation action. CAISER uses CAISER event processing language (CEPL) to represent the rules [45].

CAISER detects complex events by fusing previous and current SE data with previously detected CE data using several CE detection algorithms such as sliding window detector (SWD), weighted sub window SWD (WSD), temporally constrained template matching (TCD) and multi-layer event detector for generic application (MEGA). It also supports inter-component communication through popular social messaging apps like Telegram, Skype, Twitter and Jabber. The event pattern detection in CEP can be divided into two main methods, which are exact matching and similarity matching.

i) Exact matching: It requires all the elements in the event sequences are matched with the given event pattern template [46]. The computational cost of this approach is low. However, it is vulnerable to detection failure due to noise and system error, which commonly happen with event sequences. Clean sequences will result in high accuracy and precision, as well as quick detection and low computational costs. SWD and decision tree are examples of the exact matching approach.
ii) Similarity matching: In this approach, several criteria are used, such as Hamming distance [47] and editing operational numbers (edit, add and delete), which are performed to make both sequences similar [48]. This matching technique has been used in Non-Finite Automata (NFA) [49] and Hidden Markov Method (HMM) [50]. TCD and MEGA are examples of similarity matching approach.

Applications of CEP

There are many applications in the engineering field based on CEP. Human operators are required in traditional data analysis and mining systems to analyse massive amounts of generated data. Furthermore, the system still lacks the ability to identify unknown event patterns. As a result, the automated technique was created by combining statistical learning

Table 4 Example of available CEP engine

CEP Engine	License	Platform Development	Event processing Language (EPL)
Esper/NEsper	Open-source	C# Java	Esper EPL
Drools	Open-source	Java	Drools Rule Language (DRL)
RuleCore	Open-source	Java	RuleCore Markup Language (rCML)
Coral8	Commercial	Java	Continuous computational language (CCL)
Progress Apama	Commercial	Studio	Progress Apama EPL
StreamBase	Commercial	Studio	StreamSQL
TIBCO BusinessEvents	Commercial	Studio	TIBCO Business EPL
CAISER	Commercial	Visual Basic 6	CEPL
WSO2	Commercial	Java	WSO2 EPL
Siddhi	Open-source	Java	SiddhiQL
Apache Kafka	Open-source	Java & Scala	Kafka SQL
Google Cloud Dataflow	Commercial	MillWheel & FlumeJava	Dataflow SQL

theory algorithms and event processing systems. CEP is greatly recommended for deployment because it can obtain various sources of events, predict what will happen, and it can be used in real-time data analysis. Applications of CEP include:

i) Wireless sensor networks: The proliferation of sensor networks has increased the need for managing and detecting meaningful events (both simple and complex). There are two approaches to sensor networks: centralised and distributed processing [51, 52]. In the centralised approach, input sources are routed to the gateway for processing. However, this approach consumes more energy, and transmission latency causes event detection to be delayed. The distributed approach, on the other hand, allows for event evaluation on the node before sending the events to the gateway. These approaches decrease communication overhead while improving system performance [53, 54].
ii) Intrusion detection: The intrusion detection application is an extremely important system. The system must be able to process multiple sources of data in real-time and perform multiple actions such as notifying administrator and providing relevant data to stop the intrusion attempts. As a result, CEP methods are perfect solution for use in this type of system [55–57].
iii) Smart buildings: Smart building is a home or any building that is outfitted and integrated with surveillance and ambient devices to provide safety, security, comfort, communication, entertainment, and technical managements. CEP technology makes use of a framework infrastructure function that is suitable for smart home applications [58, 59]. Smart homes can improve safety, automate tasks, remotely monitor home activities [5], improve energy efficiency, and offer comfort to the user [60]. Smart home systems still require further investigation and development in terms of providing robust security for all connected items, as well as finding efficient ways to minimise energy consumption and taking faster and more efficient actions [16].
iv) Activity monitoring and recognition systems: CEP can be used for activity recognition to monitor and analyse CE captured by CCTV [61], or other sensors such as door sensor, temperatures sensor, etc. for different purposes such building, banks, network security, or in healthcare to monitor vital signs and patient activities [60, 62].

Integration of IoT with CEP

The integration of IoT with CEP can add up valuable features to smart server room. For example, server room which has sensors for doors, movement detectors, humidity, smoke and smart WiFi router, user could make rule patterns for the streaming event data as follow:

i) If it's daytime, the door is closed, no movement is detected and no phone device is connected to the WiFi, set the server room status to nobody.
ii) If the door is opened, sequence of movement is detected, set the status to person enter the room.
iii) If no movement is detected in the server room and the door is unlocked, then lock the door and turn on the alarm.
iv) If the temperature is high and smoke is detected, then set the status to fire is detected.
v) If it's winter, set the room temperature to 18 °C or turn off the air condition to save energy.
vi) If it's summer, lower the air condition temperature to 14 °C.

The confidence level of each event can also be taken into account if the occurrence of events does not exactly match the user rules. For example, high temperature event produces a confidence level of 25% if only one sensor out of four detects high temperature. The confidence level of fire detection event can be 60% whenever there is sensor fusion (i.e, smoke is detected, temperature is high and humidity is low). In addition, the confidence level of fire detection event can produce 100% when there are complex events (i.e. temperature is high & smoke is detected, temperature is high & humidity is low). Having a group of rules like these will add up a lot of value to smart server room. In fact, such capabilities can be easily ported to other domains such as industrial applications, for example, many machine tools could be easily controlled and maintained if certain simple rules are met such as "if the red button of a specific machine is pressed, then it must be stopped operating" [63].

Datacentre security

IoT attack surface areas show that all major components of IoT systems (i.e., IoT physical devices, communication channels and application software [33]) can be exploited and hacked. As a result, security should be the priority in building and maintaining IoT systems. In order to better integrate security into every aspect of an IoT system, security should be considered during the design phase, regardless of the system scale or type. There are four important aspects that need to be taken into consideration when designing security system for datacentre server room, which are **prevention**, **detection**, **prediction** and **response** as illustrated in Fig. 6. Security policies, access control and security awareness procedures

are all interconnected and should be designed early during the **prevention** phase. The policy should define the duties of the organization, the employees and management. Through security awareness program, employees should be educated on the importance of security, the usage of security measures, reporting procedure for security breaches and their tasks as outlined in the security policy. Access to server room should be restricted and given only to those who need to know. The organization should create user accounts by issuing identifiers, authentication methods to validate the identifiers, and authorization rules to regulate the access. Once an organization has implemented a policy, adopted an awareness training program and established access controls, it should implement detection schemes and response plans. It is important to take a proactive approach to prepare for cyber-attack or physical disaster instead of a reactive ad hoc response to an undervalued threat. The detection of a server compromise is critical. The most critical component of this strategy is the **timely detection** and warning of a compromise. For this purpose, **intrusion detection system (IDS)** is used. The systems should be capable of detecting attack signatures as well as changes in files, configurations, and any other activities. The entire server room should be monitored in order to be protected. Intrusion detection tools must be able to differentiate between normal system activities and malicious activities. The IDS is more than just an alarm; it is an alarm with intelligence, therefore, it can be trained to **predict** future events based on previously seen events. A **timely response** is required for the detection process to be useful. The response to an event should be thoroughly planned in advance. Making immediate critical decisions or creating policy during attack is a recipe for disaster. Organization should have a security operations centre (SOC) to make a response action to handle incident, for example, cut off the intruder's connection, eliminate the source of the incident, killing or restarting a process, restarting device, closing a port of device, and **recover** the affected system [64]. An intelligent and automatic response to any attack is most preferred option and it is refer to as **self-healing** mechanism for IoT devices [6]. A robust SOC should incorporate the discussed aspects to intelligently monitor and analyse activities on datacentre room.

In the next section, we will discuss the physical and cyber security related to datacentres based on IoT, followed by a comprehensive review of related works to physical security and cyber-security based on CEP and ML/DL, and the potential of integrating ML/DL with CEP is then reviewed.

Physical security

Server room issues can be catastrophic and maintaining a datacentre server room can be a major challenge. For instance, datacentre server needs the humidity and temperature to be maintained at certain range. High temperature can cause breakdown of the server, and high humidity can cause condensation and equipment failure or corrosion, while low humidity can cause electrostatic discharge, thus precision air condition to need to be controlled to maintain the humidity and temperature. Other sensors can also be needed, such as a motion sensor or camera to monitor intrude activities, smoke detector for early detection of fire, water leakage sensor to discover water leakage at early phase, AC voltage detector to monitor any surge from electrical sources, and door contact sensor to monitor unauthorized entry. A quick reporting of undesirable environmental condition to server manager can avoid heavy damage. Figure 7 shows an example of modern datacentre server with environmental monitoring system that consists of three major components: distributed sensors (e.g. door, fire, temperature, etc.), a base unit or hub, and network connectivity and integration [65]. This datacentre monitoring system can be connected to LAN/WAN network and notify the manager by SMS, Email or phone call in the event of an alarm condition. The system can also be connected to a beacon or siren for an audible alarm.

In literatures, few works that implemented IoT in datacentres. For instance, in Yamanoue et al. article [66], a method for monitoring server rooms using an IoT system was proposed. The system can configure and control terminal sensors behind a network address translation

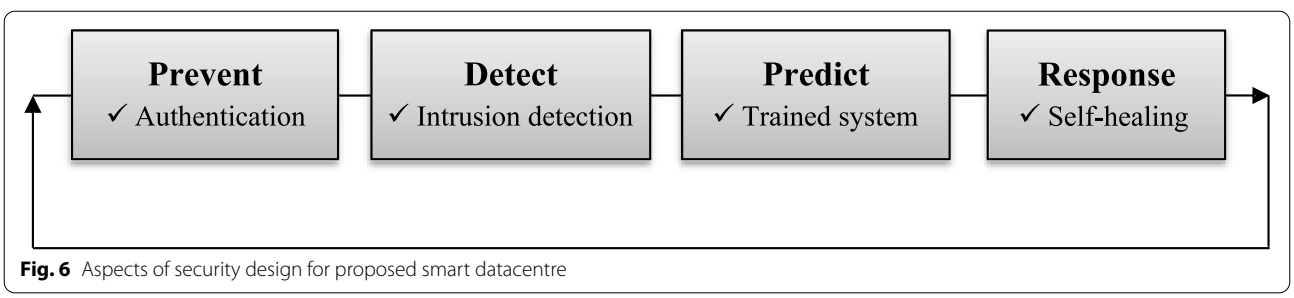
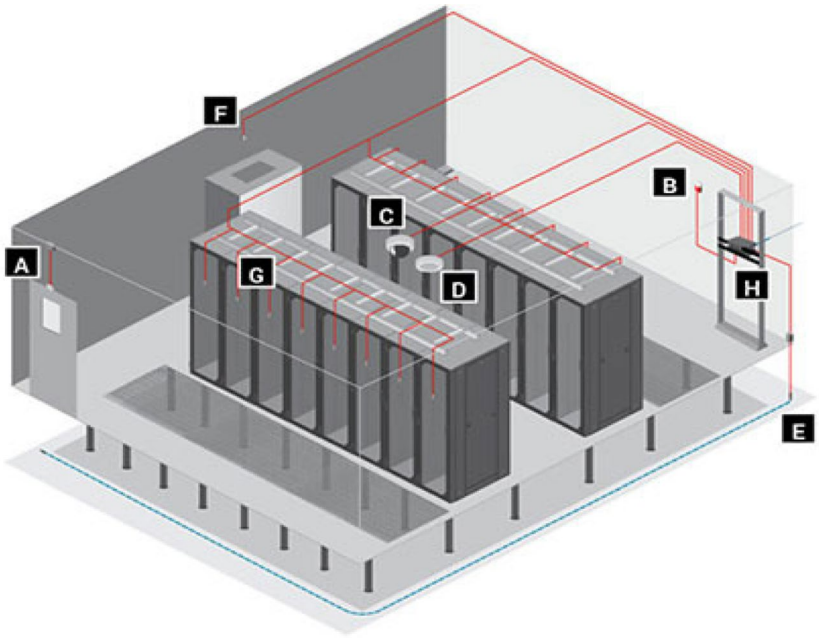


Fig. 6 Aspects of security design for proposed smart datacentre



KEY	DESCRIPTION
A	Door Sensor
B	Siren and Strobe Light
C	Camera
D	Smoke/Fire
E	Rope Water
F	Dual Temperature Humidity
G	Front-Rear Temperature String
H	Hub

Fig. 7 Example of modern datacentre server room with environmental monitoring system [65]

(NAT) router via a Wiki page or engine (such as PukiWiki [67]) over the internet. PukiWiki is software that allows users to create and update pages or entries collaboratively using a web browser. This IoT system was made up of Wiki pages and a Wiki Bot that runs on a Raspberry Pi and was equipped with several sensors. Bots are usually malicious remote-controlled computers or programs that form a botnet, but here the authors used them for beneficial tasks [68]. Their system was successfully used to monitor and detect air conditioner malfunction in a server room, however, the system requires 24 Wiki pages for each hour and 31 Wiki pages for each day in a month which causes challenging for debugging the Wiki IoT system when error is occurred. However, an intelligent system is preferred for such a case to handle the monitoring without human intervention. CEP technology with the integration of ML can provide a notable solution.

Complex event processing (CEP) systems have become increasingly important due to the recent development in communication and integrated circuit technology. Arduino can be used as an IoT device to create a variety of smart space systems. These systems can be used in a variety of applications such as industries, greenhouses, and smart homes. In Kaya et al. article [69], CEP using IoT devices based on Arduino board is proposed. Several sensors are connected to an Arduino board where a sensor node is created using the ESP8266 WiFi module for communication and to collect necessary data. Sensors such as light, humidity, temperature, sound, soil moisture rain and snow are used in their study to create a smart environment. The sensor data is transmitted to a central server over the internet via an access point. On the central server, an event processing system based on a CEP tool is created, and a publish-subscribe event-based system is implemented. The collected sensor data must be processed online to determine whether an event occurred or not based on predefined rules. When an event occurs that is related to a specific topic, an alarm notification is sent to mobile users with the event information so that necessary precautions can be taken.

Table 5 summarizes the existing research papers that used CEP for physical security. In Shahad et al. article [45], CEP engine based on CAISER and NEsper are used to monitor and analyse complex event produced by CCTV and door sensors for activity recognition in smart building applications. CIASER with TCD complex detection method shows the best detector with less time latency, higher precision value and accuracy compared with SWD, WSD and NEsper. TCD is better because its parameter is easy to vary based on the desired scenario. NEsper is also good for complex event detection, but it is limited to exact matching approach and its parameter is restricted.

Suspicious loitering is one of the anomalous activities that precede unwanted events such as break-ins, burglaries, and robberies. In [61], suspicious loitering detection from annotated CCTV is proposed based on CEP approach. Complex event processor for scientific and

Table 5 Summary of existing works on using CEP for physical security purposes

Paper	Year	Threat	Methodology	Pros (+) & Cons (−)
[62] Itria et al.	2014	Armed robbery, act of vandalism, dangerous object placed by person, melee.	Method for detecting critical situations that employs event correlation technologies to perform online analysis of real-world event via CEP engine such as Esper is proposed.	+ Event correlation technology is used to perform online analysis of real-world events via Esper. - Rules are manually defined for each case.
[45] Shahad et al	2018	Anomalous activities in smart building	CEP engine based on CAISER and NEsper are used to monitor and analyse complex event produced by CCTV and door sensors for activity recognition in smart building applications.	+ Comparison of two CEP engines, and different CE detectors using metrics of confusion matrix. - Only person movement activity is considered, and no standard/public dataset was used.
[61] Shahad et al.	2018	Break-ins, burglaries, and robberies	Suspicious loitering detection from annotated CCTV is proposed based on CEP approach (CAISER)	+ Performance of different CE detectors was evaluated with noisy and no-noise event annotation. - Not implemented for real-time smart surveillance system.
[70] Amru et al.	2018	Loitering ambient, falling and injured	Real-time event detection based on CEP (CAISER) for intelligent building surveillance system is proposed.	+ System was evaluated at different alignment situations in term of detection accuracy and ROC curve. - Limited to single person tracking.
[71] Mijović et al.	2019	Fire or terrorist attack	Intelligent and event driven layer powered by CEP is proposed to handle a flood of information coming from many sensors in critical infrastructures such as airports.	+ System allows operator to react automatically or manually to risk. - No detail analysis of system performance in term of accuracy or comparison with different systems.

engineering application platform known as CAISER is used as a CEP engine. The collected input data is annotated using InViSS. Annotator software. Four different types of similarity search-based event detectors, namely, Sliding Window Detector (SWD), Weighted Sliding Window Detector (WSWD), Temporally Constrained Template Match Detector (TCD) and Multi-Layered Event Detector for General Application (MEGA) are used and evaluated under different added noise. The study showed that the best performance is achieved by MEGA in terms of average accuracy and ROC curve. Similarly in [70], real-time event detection based on CEP for an intelligent building surveillance system is proposed to assist the intrusion detection for security reliability.

Critical infrastructures (CI) such as airports are difficult to handle during an emergency (i.e., fire or terrorist attack) due to their complication, big size and number of stakeholders involved. Intelligent and event driven layer powered by CEP is proposed in [71] to handle a huge of information coming from many sensors and legacy monitoring systems. The system aims to provide recommendation and decision support to CI operators.

In [62], CEP approach for crisis management system is suggested. The paper describes a method for detecting critical situations that employs event correlation technologies to perform online analysis of real-world events via CEP architecture such as Esper. Event correlation is a technique for detecting patterns and situations of interest in the emergency management context by correlating events gathered from various sources, including crowd sensing and crowd sourcing. World-heritage protection scenario was selected as case study to detect any dangerous events such as acts of vandalism, armed robberies, attack organization, placing of dangerous objects near monuments and recognition of keywords in file audio such as weapon, bomb and so on. The purpose of this approach is to handle huge amount of incoming information, detect critical situation in progress before they happen and provide early warning in order to make preventive action.

Cyber security

A widely used technique for cyber security is the intrusion detection system (IDS). An ID is a technique that has the ability to find out any suspicious activities or any breaches in the network security. An IDS should then react to the intruders or hackers and send alerts to the administrator. Generally, IDS can be divided into two categories namely signature-based techniques and anomaly-based detection techniques. Signature-based techniques are vulnerable to zero-day attacks and necessitate the constant updating of the signatures database [39]. While, anomaly-based detection techniques, seek to identify abnormalities in network traffic or system behaviour that could indicate the existence of malicious activity [72]. The anomalies could be created due to many factors, such as noise or phenomena that has a certain probability of being happened by certain conditions, which threatens the confidentiality, availability, and integration of data for an organization [73, 74]. Therefore, anomalies are unusual behaviours triggered by a person/object that leave footsteps in the computing environment [75]. These footsteps are tracked in order to recognize attacks, particularly those that are unknown. The anomaly-based system detects any deviations from normal behaviour in the computing environment by continuously updating a model of normal behaviour in the desired environment based on normal data acquired from known users [76]. Anomalies are divided into three categories, point, contextual and collective anomalies [72]. Anomaly detection can find applications in a variety of domains such as physical or cyber intrusion detection, fault detection, event detection in sensor networks, fraud detection, ecosystem disturbances and health monitoring systems [4, 19]. There are several methods that can be used for an anomaly-based detection system, which are well-discussed in the literature [4], and they can be summarized as follows:

i) Data mining: This technique aims to extract information from large datasets, in similar way to extract gold from massive rocks and sand [77]. The derived information is identified as useful patterns in the data [78]. One of the benefits of this approach is the capability to automatically create models based on the traffic description. The data mining approach is suitable for an unbounded, continuous, and promptly increasing online data stream [79]. In the design of an anomaly-based detection system on this approach, procedures consisting of association rule phase, clustering phase, classification and regression phase, and outlier analysis phase are commonly used [59, 77]. There are a variety of algorithms that can deal with each phase such as K-means algorithm for clustering, K-nearest neighbour algorithm for classification.
ii) Machine learning: This approach depends on two phases: the training or learning phase and the testing or detection phase [80]. The training phase is based on mathematical functions or algorithms that learn the features of the desired environment by using normal data as a reference input. These features are then employed for detection and classification in the second phase [81]. Supervised learning is a type of ML technique in which the training dataset's properties are used to develop a classification model in the first phase, which is subsequently used to categorise new

unknown instances [82]. Unsupervised learning is a ML technique that relies on the data's features rather than clustering training data [82]. Examples of supervised techniques are support vector machine (SVM), deep neural network (DNN), convolutional neural network (CNN), recurrent neural network, unsupervised techniques are autoencoder (AE), generative adversarial network (GAN).

iii) Statistical-based model: this approach is based on statistical processes (e.g., Gaussian distribution) [83, 84]. The statistics of historical user behaviour are utilised to develop a normal model, which is then used to detect any deviations from the model. These discrepancies are referred to as abnormal data. The statistical model technique detects irregular traffic from observed traffic patterns by applying statistical mathematical operations to a training dataset [85].

iv) Rule-based model: this technique is based on the formation of rules for the desired environment. These rules are derived from the patterns of data traffic. In a rule-based detection system, any anomalous data flow that violates these rules is detected and treated as an attack [86, 87]. The rule-making process is influenced by previous system behaviour. Therefore, to prevent an unreasonably high false positive rate (FPR), the ambient environment must be monitored for a long period to provide sufficient data [4]. Fuzzy rule based detection system is an example of this approach. Table 6 summarizes the advantages and disadvantages of the discussed techniques.

Related papers based on CEP

Table 7 summarizes the existing research papers that used CEP for cyber security. In [56] Jun and Chi proposed the integration of IDS with CEP to quickly react to any hacking attacks and malicious activities in IoT network environments. The proposed system uses rule model approach to deal with different patterns in events and to process a large volume of messages with short latency. The implementation for real-time event processing is developed by CEP engine called Esper to detect LAND (local area network denial) attack which is DoS attack consists of sending a special poison spoofed packet to host device, causing it to lock up. Figure 8 depicts the architecture of integrated IDS-CEP system. The system collects raw data (network traffic and event usage) from IoT devices. This data need to be cleaned and filtered by event filter since they are collected from network card and they usually are inconsistent and duplicated. The filtering process will remove noisy and redundant data, and pre-process them into unified format. The data will then be forward to event database (event recording module) and complex event processor. The latter, whose core module is Event Processing Repository (EPR), is responsible for detecting security events. Event processing model (EPM) is stored in EPM, and it is a collection of correlations of events. New events will be generated when the input data stream matches the EPM constraints. System administrator can define the security events based on specific requirements. The authors used Esper as CEP engine to react to different security events and identify the requirements of application. The action engine is responsible for events triggered by EPM matching, for example for security event with low level of danger, it stores the log file for intrusion activity, and for high level security event, it adds the source IP address into blacklist and cuts off the TCP links to relevant IoT service. Their suggested approach is CPU intensive; however it consumes less memory and shows effective real-time performance. The main characteristics of this CEP-based system are that it works smoothly in real time and performs well in detecting abnormalities in an IoT system utilising an event-processing mechanism [56].

In [92], a hybrid IDS and CEP is proposed to deal with data leakage and network breach such as multiple root accounts, shared root access, top priority privileges and shared file system privileges. The system is called hybrid

Table 6 Merits and demerits of anomaly-based detection techniques

Approach	Advantages	Disadvantages
Data mining	+ The models are automatically generated. + It can be used in variety of situations. + Appropriate for use with online datasets.	- It depends on historical data. - It is based on complex algorithms.
Machine learning	+ High accurate detection. + It can be used to handle large amounts of data.	- Requires training data. - Extensive training time
Statistical-based model	+ Appropriate for use with online datasets. + Simplicity of the system.	- It is based on previous behaviour. - The accuracy of detection is determined by statistical processes.
Rule-based model	+ It is appropriate for online datasets. + Simplicity of the system.	- System is based on a set of rules. - There is a high risk of false positives rate.

Table 7 Summary of existing works on using CEP for cyber security purposes

Paper	Year	Threat	Methodology	Pros (+) & Cons (−)
[88] Aniello et al.	2011	Stealthy port scan	Architecture of IDS which uses CEP engine (Esper) is proposed. An algorithm for detecting SYN port scan to check for any malicious behaviour of host activities is implemented.	+ High detection and low false positive rates were achieved. - Only one type of cyber attack was considered.
[89] Cheng et al.	2011	Network service	NEPnet monitoring system which can process events for anomaly detection is proposed. NEPnet is built on CEP to support variety of event correlations by creating a tree-based monitoring net for anomaly detection.	+ NEPnet can detect anomaly with high speed compared to Esper CEP engine. - Requires pre-defined rules.
[90] Gad et al.	2013	DoS attack, SYN flooding	CEP (Esper) is used for network analysis and surveillance to effectively filter undesired information and infer high-level information.	+ Possibility to effectively deduce meaningful high-level data in network analysis and surveillance with means of CEP. - No analysis for accuracy of intrusion detection.
[56] Jun et al.	2014	LAND attack	Integration of IDS with CEP (Esper) to quickly react to any hacking attacks and malicious activities in IoT network environments.	+ Works smoothly in real time and performs well in detecting abnormalities. - No analysis for accuracy of intrusion detection.
[91] Jayan et al.	2014	DoS attack, buffer overflow	Method for pre-processing a vast input for CEP engine is described. The method aims to extract only relevant data that concerned the CEP engine (Esper).	+ CEP rules are built based on risk taxonomy to find the attack patterns. - The rules can be very long.
[92] Mohan et al.	2015	SYN flood, port Scans, LAND, NTP flood, Back, Neptune, POD, Smurf and Teardrop attacks.	Hybrid IDS and CEP by integrating the output of host IDS and network IDS into the CEP module. The host and network features are dynamically updated when a threshold is crossed. KDD99 dataset is used for testing the system.	+ System is evaluated in term of CPU, RAM usage and compared with other works in term of detection rate. - Sophisticated set of pre-defined rules are used.
[93] Vegh et al.	2016	Private keys, authentication	System that provides hierarchical access to data via a digital signature algorithm is proposed. The system aims to detect and prevent attacks using WSO2 CEP.	+ Digital signatures for access control are used by the model. - No performance evaluation of the proposed system is conducted.
[57] Cardoso et al.	2018	UDP flood, SYN flood ICMP flood and ports scan attacks	DDoS detection system (called CEPIDS) capable of identifying malicious traffic in real-time IoT environments using CEP rules is proposed and tested using Raspberry Pi.	+ Tested on a Raspberry Pi 3, achieved good accuracy in detecting attacks, good CPU and RAM usage to enable it for IoT compared with Bro and Snort IDSs. - Higher lost packet rate compared to Snort IDS, but lower than Bros IDS.
[94] Devi et al.	2021	TCP SYN ACK, TCP SYN flood, LAND, ICMP and UDP floods attacks	Cloud-based DDoS detection and defence system by CEP is proposed to handle traffic from various attack sources and correlate the event patterns with real-time traffic in order to protect the cloud from DDoS attacks.	+ System achieved high detection accuracy, and defense system uses dropping attack traffic IPs as remediation action to protect the cloud from any DDoS attacks. - Detection is limited to known attack pattern only.
[55] Lima et al.	2022	SYN flood attack (TCP), and denial of service attack	Intrusion detection and prevention system (IDPS) based on CEP (Esper) called Beholder is proposed and used for IoT applications that use MQTT and CoAP protocols. The work uses CEP technology to process messages exchanged between IoT devices in order to identify patterns that could be used in a cyber attack.	+ The performance of the system is compared with Snort IDS. - Only few attacks are considered, and new CEP rules to detect other types of attacks in MQTT and CoAP applications are needed.

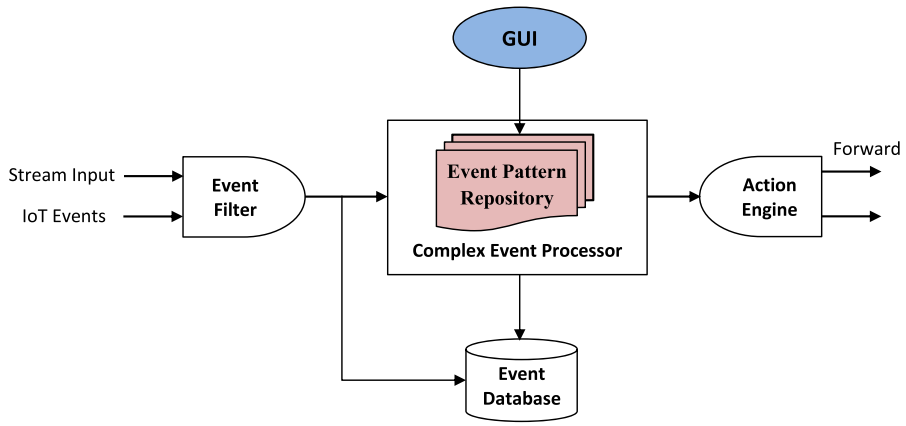


Fig. 8 CEP-based IDS architecture for IoT [56]

IDS since the authors integrate the output of host IDS and network IDS into the CEP module. The host and network features are dynamically updated when a threshold is crossed. This is done to reduce the amount of IDS traffic in the network. Multivariate Correlation Analysis (MCA) is employed as CEP module to estimate and characterise the network's normal behaviour, and the resulted status are sent to the rule engine, which notifies the administrator in the case of any anomaly from the normal pattern. The suggested system with MCA technique achieved good detection rate with comparison to Kmeans+KNN and biological memory cell techniques using KDD99 dataset.

In [94], cloud-based distributed DoS (DDoS) detection and defence system supported by CEP is proposed to handle traffic from various attack sources and correlate the event patterns with real-time traffic in order to defend the cloud from DDoS attacks. The system is used to detect different kinds of DDoS attacks, for instance, TCP SYN ACK, TCP SYN flood, LAND, ICMP and UDP floods by correlating with cloud resources such as source IP address, destination IP address, port and subnet. The defence system uses enforcement policies such as dropping attack traffic IPs as remediation action to protect the cloud from any DDoS attacks. The system provides high accuracy of detection for flooding attacks in transport and network layers if it is compared to non-CEP based architectures, nevertheless, the detection of unknown patterns resulting in zero-day attacks, is considered a drawback of their work, which could be addressed by integrating ML-based models.

In [57], Cardoso et al. proposed DDoS-based detection system (called CEPIDS) capable of identifying malicious traffic in real-time IoT environment using CEP rules, which is tested on a Raspberry Pi. The findings suggest that the proposed system can be executed on IoT devices with low processing power, and that CEP is a viable method for improving real-time detection of DDoS attacks in IoT contexts. The system achieved good accuracy in detecting SYN flood, UDP flood, ICMP flood and port scan attacks. The CEP based IDS is also achieved good CPU usage, RAM usage and packet loss rate compared with Bro and Snort IDSs.

Data processing in event-driven architecture [95] is triggered immediately upon the arrival of new data. However, the event-driven paradigm is not sufficient alone to identify and solve critical situation. Hence, a mechanism for effectively filtering undesired information and inferring high-level information is required. Therefore, in [90], CEP is used for network analysis and surveillance that includes detection of intrusions and network monitoring, congestion, misconfiguration or faults situations. Different event patterns were written in Esper EPL languages for detecting DoS attack such as SYN flooding. The data samples have been collected in real networking scenario using Wireshark.

In [93], a system that provides hierarchical access to data via a digital signature algorithm is proposed. The system aims to detect and prevent attacks using WSO2 CEP. Any action within the cyber-physical system, such as digitally signing a message, can be considered an event. With this in mind, the authors propose a solution that enables the creation of a secure cyber-physical system with effective attack detection architecture.

A method for pre-processing a vast input for CEP engine is described in [91]. The method aims to extract only related data that concerned the CEP engine. The pre-processing module involves four sub-modules which are data reader, raw data parser, log evaluator and data writer. Esper is used as CEP engine to store risk rules

and to detect the severity of threat. Risk classification is constructed based on three levels, where any security threat (i.e. DoS attack, buffer overflow) is assigned alarm ID. Architecture of intrusion detection system which uses CEP engine Esper is proposed in [88]. The authors implemented an algorithm for detecting SYN port scan to check for any malicious behaviour of host activities.

In [89], a scalable NEPnet monitoring system which can process events for anomaly detection is proposed. NEPnet is built on CEP to support a variety of event correlations by creating a tree-based monitoring net for anomaly detection. NEPnet can be easily used by network services due to its scalable SQL-like language, which supports various temporal relationships such as negation, aggregate functions (i.e., min, max, sum, avg), conjunction and disjunction. Preliminary experiments demonstrated that NEPnet can analyse event correlations and detect anomalies in network services more efficiently than Esper engine in terms of CPU and memory utilizations [89]. Lima et al. in [55] proposed intrusion detection and prevention system (IDPS) based on CEP (Esper) called Beholder, and used for IoT applications that use MQTT (Message Queue Telemetry Transport) protocol and CoAP (Constrained Application Protocol). The work uses CEP technology to process messages exchanged between IoT devices in order to identify patterns that could be used for attacks. To evaluate the performance of their proposed work, SYN flood attack and denial of service attack are considered for applications that use MQTT protocol and error flood attack for CoAP applications. The results showed excellent detection rates performed by Beholder in terms of accuracy, precision, recall and F1-score metrics.

Related papers based on ML and DL

ML and DL fall under the artificial intelligence (AI) domain and aim to extract meaningful information from large amounts of data [96]. Due to the invention of extremely powerful graphics processor units (GPUs), these techniques have gained significant prominence in the domain of network security during the last decade [21, 97]. Both ML and DL are effective methods for extracting significant information from network traffic and forecasting normal and abnormal actions based on the patterns learned. To learn relevant information from network traffic, the ML-based intrusion detection system (IDS) mainly relies on feature engineering [98]. DL-based IDS, on the other hand, do not rely on it and is capable of automatically learns complicated features from raw data due to its deep structure [99]. In this section, we review few recent literature papers that discuss about cyber security based on IDS using ML and DL. Extensive review can be found in many surveys such as [4, 19–22].

In Jithu et al. paper [100], intrusion detection system to uncover DDoS Botnet attack for IoT is proposed using deep neural network (DNN). The authors used asynchronous security scans to analyse traffic patterns on various types of IoT devices. A BoT-IoT dataset is developed based on network environment developed in the Cyber Range Lab of the centre of University of New South Wales (UNSW) Canberra Cyber. The evaluation of the anomaly-based detection system is done using different metrics such as accuracy, precision AUC-ROC and F1-score.

To detect any anomalous behaviours in IoT network, Anthi et al. [101] constructed a simulated smart home environment and used ML techniques such as Naive-Bayes, SVM, decision tree, and random forest. They used their model to predict several attacks strategies such as reconnaissance, DoS/DDoS, and spoofing, and distinguish whether network activity is malicious or benign. In [102], a multi-class classifier is used to detect any unauthorized IoT devices that are not within a given white list. A simulated environment is generated, and essential features are extracted using supervised machine learning approach such as Random forests. One disadvantage of [101, 102] is that they both employ simulated data, making the model more environment-specific. Doshi et al. [103] tested five ML algorithms such as K-nearest neighbours (KNN), support vector machine with kernel (LSSVM), decision tree (DT), random forest (RF) and neural network (NN) to detect DDoS attacks in IoT networks. They showed how exploiting IoT-specific network behaviour to update feature selection can lead to high accuracy in DDoS detection for IoT network traffic using a range of ML algorithms. Different metrics are used to evaluate the algorithms such as precision, recall, F1-score and accuracy.

The authors of [104] presented an anomaly detection system that employs network behaviour snapshots and deep-encoders to detect any abnormalities network traffic originating from compromised IoT devices. They did this by creating an IoT network and infecting it with Mirai and Gafgyt/BASHLITE botnets. They generate a deep autoencoder (AE) for each device separately based on the features accumulated through source/destination IP addresses and MAC addresses.

Meanwhile, Yuan et al. in [105] proposed a deep learning-based defence method called DeepDefense to detect DDoS attacks. Their intrusion detection system is built based on convolutional neural network (CNN), recurrent neural network (RNN) model such as long short-term memory network (LTSM) or gated recurrent unit neural network (GRU). The system was trained using the ISCX2012 dataset. Their approach has proved to lower error rates compared to other machine learning such

as random forest, however one disadvantage is that the dataset is outdated and not well updated.

Ibitoye et al. [106] used Self-normalizing Neural Network (SNN), a variant of forward Neural Networks (FNN), to build an anomaly detection module for classifying intrusion attacks in an IoT network. They tested their model using BoT-IoT dataset developed by Cyber Range Lab (CRL) of the center of UNSW Canberra Cyber. The results showed that FFN outperform the SNN for intrusion detection in IoT network, however, SNN has greater flexibility against the adversarial samples from the IoT dataset compared to FNN.

Thamilarasu et al. [107] created a ML-based anomaly detection (MLAD) module based on deep belief network (DBN) to detect any anomalous behaviour on insecure IoT network. The performance of their system is tested it on their own test bed for five different attack strategies such as DDoS attack, opportunistic service attack, blackhole, wormhole and sinkhole attacks. Table 8 summarizes few recent studies that used ML or DL for physical and cyber security. It is important to mention that, this section is provided to highlight how significantly ML/DL has been used for IoT cyber security purpose in literature. Comprehensive review can be found in other survey articles such as [4, 19–22].

Once attack has been detected or predicted, another important challenge remains with mitigating the cyber-attack on IoT devices. It is not feasible to manually manage alerts from millions of IoT devices or to expect clients to handle the alerts on their own. As a result, an automated self-healing mechanism is preferred to manage alerts and restore the devices to their normal, uncompromised state.

Amit et al. [6], proposed a self-healing mechanism for IoT device based on deep learning to recover from malfunction and cyber-attack (i.e., DoS or DDoS) automatically in real-time by selecting a proper remediation without the need for human interaction. Once an alert is supplied by host-based IDS (HISD), their framework supports self-healing by choosing and triggering the best remediation method as illustrated in Fig. 9. A health-monitoring module (HMM) is responsible for checking the state health (i.e., normal or abnormal) of IoT device. It continuously (or triggered once needed) collects and analyses information from a variety of sources, including disk space, memory access and utilisation, CPU usage and network connection metrics. These features are given to anomaly-detection model to check whether the system is in healthy state or not. The anomaly-detection model can be designed based on neural networks such as autoencoder or based on predefined rules method that does not have major computational complexity. Auto-remediation module acts as device's doctor, is implemented using LSTM network to choice remediation actions. It also collaborates with other similar devices on the network to get the best suitable remediation to the particular type of IoT device. The remediation procedure will be given a score by the health-monitoring module, which will represent its efficacy in treating the device. These scores are provided by evolutionary algorithm (i.e., genetic algorithm) in order to improve the remediation selection process. To assess their proposed self-healing framework, a testbed involves 35 Raspberry Pi connected to a LAN network, and CIoTA (collaborative IoT anomaly) detection framework as HISD module are used. Several DoS and DDoS attacks are simulated with suggested remediation as shown in Fig. 9. The findings showed that, the self-healing framework has the capability to learn different remediation patterns and mitigate sophisticated cyber-attacks using deep learning for IoT devices.

Table 8 Summary of selected works on using ML/DL for cyber security without CEP technology

Paper	Year	Threat	Techniques	Dataset
[102] Meidan et al.	2017	Unauthorized IoT	Random forests	Simulation
[105] Yuan et al.	2017	DDoS attack	CNN, RNN models such as LTSM or GRU.	ISCX2012
[104] Meidan et al.	2018	Abnormalities network traffic from IoT devices	Deep autoencoder (AE)	N-BaIoT
[105] Ibitoye et al.	2019	DDoS attack	Self-normalizing Neural Network (SNN)	BoT-IoT UNSW
[101] Anthi et al.	2019	Reconnaissance, DoS/DDoS, and spoofing	Naive-Bayes, SVM, decision tree, and random forest.	Simulation
[107] Thamilarasu et al.	2019	DDoS attack, opportunistic service attack, blackhole, wormhole and sinkhole attacks.	Deep belief network (DBN),	Own testbed
[100] Jithu et al.	2021	DDoS attack	Deep neural network (DNN)	BoT-IoT UNSW

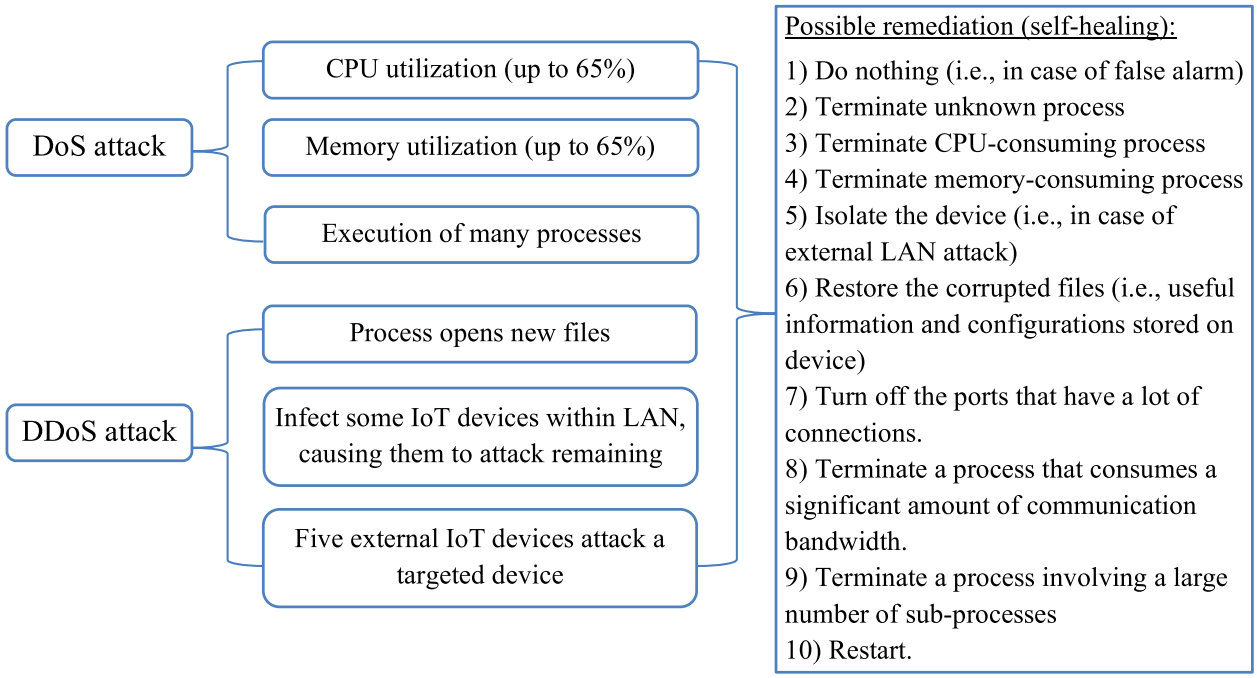


Fig. 9 Cyber-attacks and possible remediation for self-healing mechanism proposed in [6]

Integration of CEP and ML/DL

All of the above research works demonstrate how machine learning and deep learning have become an essential component of cyber security for detecting IoT botnet attacks. The process of identifying complex patterns in CEP faces with a number of challenges such as the manual and complexity of writing pattern rules and the requirement to get and process background data while considering the event stream's real-time constraints. According to Alakari et al. [108], three major obstacles must be overcome in order to develop an efficient rule mining method for fine-tuning the CEP pattern to identify situation of interest

1) The CEP pattern rules should be inferred using the user's selected context and the event stream's history.
2) The refining process should utilise the smallest amount of rules possible to prevent the issue of pattern complexity.
3) The refinement task should be completed in very short time in order to respond to new events.

The combination of CEP and ML techniques has the potential to produce improved results in term of accuracy robustness in the field of intrusion detection.

Margara et al. [109] discussed the issue of writing CEP rules manually and proposed an iCEP framework to tackle this challenge. The iCEP learns hidden causality between the received events and situations to recognise from historical traces, and use them to automatically create CEP rules. Ad-hoc learning algorithm is used to enhance the readability of events patterns. They evaluated their proposed system in terms of recall and precision using man-made and real datasets from a bus traffic monitoring scenario. But, the extracted rules are static and difficult in handling uncertainty and editing by user [110].

Since the IoT systems that design based on CEP is relied on determination of rule patterns, Mehdiyev et al. [111] proposed to use ML techniques to replace the manual identification of rule patterns. Once a pre-processing stage which deals with missing values and outliers values is completed, various rule-based data mining techniques such as One-R (One rule) [112], PIPPER (Repeated Incremental Pruning to Produce Error Reduction) [113], PART (partial decision trees) [114], NNge (Non-Nested Generalized Exemplars) [115], RIDOR (Ripple-Down Rules) [116] and DTNB (Decision Tables and Naïve Bayes) [117] available from WEKA tool were applied to detect complex events. The highest accuracy is 93.14% achieved by PART algorithm, and the worst performance is 79.89% given by ONE-R algorithm. However, the system is tested for daily routine movement activities such as walking, sitting, standing, descending or ascending stairs, and jogging.

Mousheimish et al. [118], a data mining based method is proposed to learn predictive CEP rules automatically from multivariate time series. CEP engine such as ESPER is used, and Shapelets [119] algorithm is used to achieve early classification, and extract shapelets using USE and SEE techniques then transformed them into CEP rules by autoCEP. To evaluate their system, performance metrics such as accuracy, F1 score, earliness and applicability are used for three datasets namely Wafer (predict failures in wafer manufacturing), ECG (predict if ECG data is normal or not) [120] and Robots (predict robot failure) [121].

Lee et al. [122] proposed a framework called sequence clustering-based automated rule generation (SCARG) to automatically create rules. At first, K-nearest neighbour (KNN) method is used to categorise event items, then event's similarities and ordering are compared using similarity-based clustering. Finally, to extract the complex event patterns, probabilistic graphical modelling based on Markov model is applied. To assess the effectiveness of their method, the model is employed for stock trade system and Apache Tomcat CEP engine. They compared between dynamic CEP and proposed adaptive CEP methods in terms of rate of return. According to the findings, the adaptive CEP has improved the performance metric, and it was able to effectively extract rules using historical data collected by domain expert.

Roldan et al. [123] proposed integration of CEP and ML for detecting different types of IoT security attacks by constructing event patterns whose criteria rely on the data of the network packets that predicated by ML techniques. Enterprise service bus (ESB) as middleware is used to establish data connection between IoT network and both the CEP engine and ML techniques using MEdit4CEP model-driven approach [124]. The intelligent architecture is applied to hospital IoT network to detect attacks such as UDP, TCP and Xmas post scans, and DoS attack made by malicious device. The advantage of this approach is that, the CEP engine can swiftly adapt to new scenarios if the network environment changes. Another important feature is the non-expert user can graphically identifies which security attack types must be analysed and stopped without the need to have advanced knowledge about ML, CEP or IoT networks. Supervised machine learning approach such as linear regression or support vector regression (SVR) is used in their study to calculate expected pattern values that enable the new attack detection; however, comparison to other ML algorithms is still needed.

The previous studies [111, 118] deal with IoT labelled data, however, actual IoT stream data has no labelling information and need to be trained by unsupervised approaches. Recently, Simsek et al. in [110], proposed a system named automatic rule extraction for CEP (ARECEP) to extract CEP rules from unlabelled IoT data based on DL and rule mining techniques. The framework consists of two phases which are data labelling phase and automatic rules extraction phase as depicted in Fig. 10. In the first phase, DL method such as autoencoder, CNN, RNN, LSTM, CNN-LSTM or GRU are used, and in the second phase rule based data mining method such as PART, decision table (DT), JRIP (Extended repeated incremental pruning), ONE-R, RIDOR, NNge or FURIA (Fuzzy unordered rule induction algorithm) are used. The data coming from environmental sensors are sent to both of the CEP engine and historical database (HDb). The CEP engine only able to detect data with predefined rules, however, the whole data is stored in HDb to be used later for rule extraction purpose. In their proposed framework, the data in the HDb is trained by DL algorithms to label theses data as normal or anomalous in the first phase. The anomalous data are converted into rules by using rule-based mining methods in the second phase. The extracted rules are analogous to if-then-else rules, which are transformed into CEP rule syntax and stored into CEP database for further investigation. Air pollution dataset is used to evaluate their proposed system. The performance of the first phase is assessed by the reconstruction error, which is the difference between the input vector and output vector. Other performance metrics such as recall, precision, F1-score, accuracy and ROC area are also used to evaluate the success rates of the rule-based data mining algorithms. Their proposed framework can be further investigated on cyber security dataset and enhanced where it is appropriate. Table 9 summarizes the existing studies that integrated CEP with ML/DL for physical and cyber security.

Xi et al. [125] proposed counter-terrorism early warning system for the urban environment based on IoT technology to identify suspicious activities, behaviours, items, persons, and vehicles by comparing historical data of terrorist attacks with massive real-time information. The framework consists of three layers: intelligence perception, intelligence identification and intelligence inference. The intelligence perception layer was used to collect information from smart sensors installed in different areas of cities, and the intelligence identification layer was implemented to identify features of terrorist activities based on event-based and semantic-based terrorist action. The intelligence inference layer was designed by combining CEP with ML to provide timely response and awareness of potential threats.

Roldán et al. [126] integrated CEP with ML for cyber attack pattern recognition in IoT. The framework used ML method to enable automatic generation of CEP patterns from categorized or uncategorised data, depending on whether the goal is to classify attacks

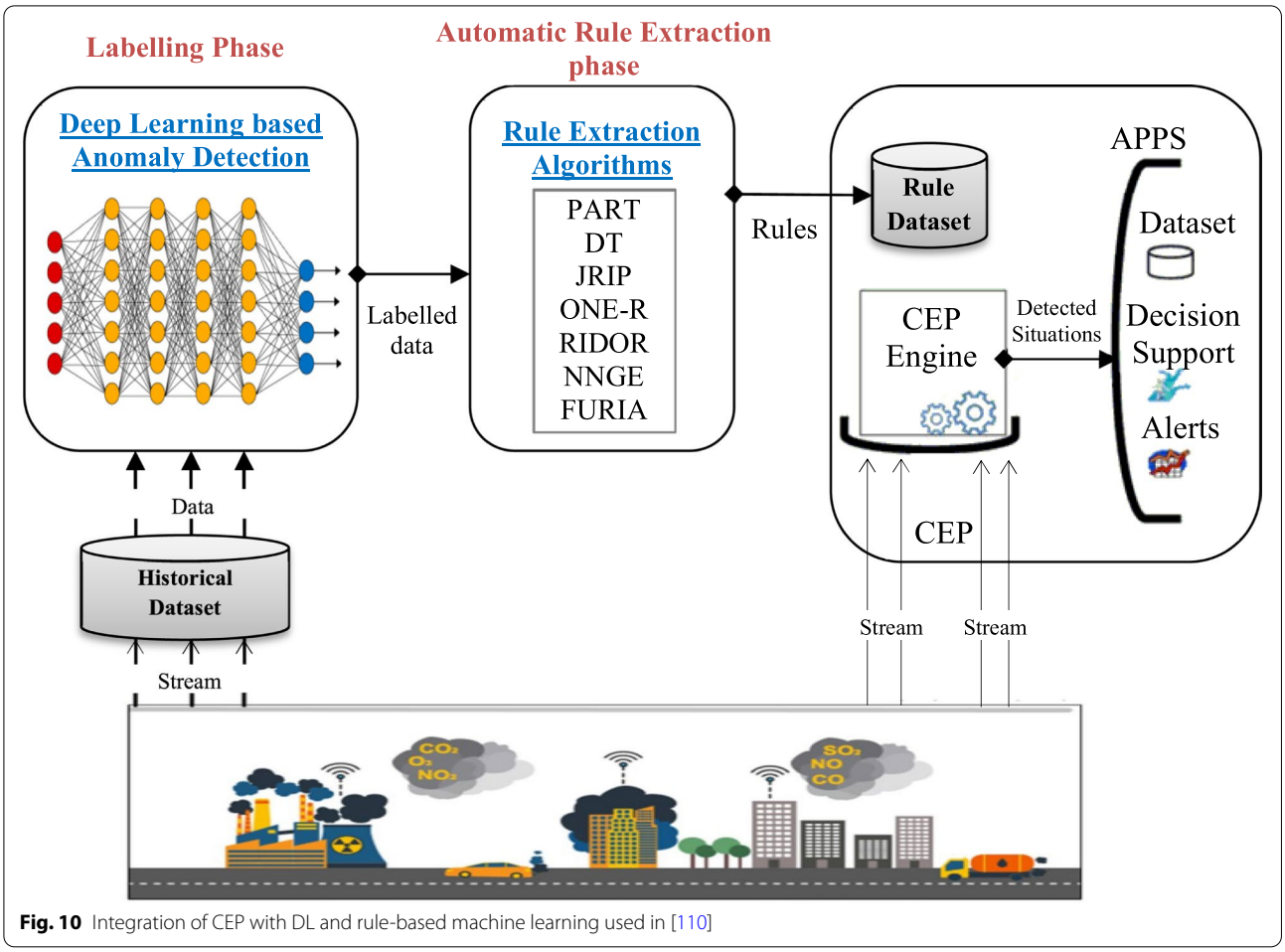


Fig. 10 Integration of CEP with DL and rule-based machine learning used in [110]

or identify anomalies. Siddhi engine is used to generate patterns automatically based on defined equation contains parameters such as threshold, means variance explained of components obtained by PCA. Dataset contains attacks such as subscription fuzzing, disconnection wave, TCP syn scan, UDP scan, Xmas scan and Telnet connection, was extracted from a network using MQTT and used to show how the proposed system works in environment with different attacks. In addition, a comparison between two CEP engines namely Esper and Siddhi (WSO2) is conducted in [127] to ensure that the designing of the architecture integrated CEP with ML may be implemented with various vendor technologies while preserving its proper functioning and that it is not necessarily bound to specific vendor technologies.

Challenges and recommendations

The CEP-based intrusion detection system is an attractive technology due to its flexibility (i.e. not bound to a specific application domain), and simplicity to adjust its rule patterns based on different intrusion strategies. However, there are multiple research gaps that concern CEP interoperability, uncertainty query language and real-time response requirements as illustrated in Fig. 11 due to massive deployment of IoT devices in future applications.

These challenges with suggested recommendations are highlighted as follow:

- There is currently no standard method for defining events, event pattern types and rule operators, and consequently no unified approach for exchange of event pattern instances between different CEP systems. This causes issues when several big data, analytics solutions and techniques are needed to be linked together [128]. An open approach with the capability to leverage current technologies such as R [129], web services, cloud analytics from Azure [130] and REST [131] will enhance the CEP technology.
- In large-scale IoT applications, existing CEP technology faces the challenge of massive distributed data,

Table 9 Summary of existing works on integrating CEP with ML/DL

Paper	Year	Threat type	Applied for	Methodology	Techniques used
[109] Margara et al.	2014	None	Bus traffic monitoring scenario	iCEP framework is proposed to learn hidden causality between the received events and situations to recognise from historical traces, and use them to automatically create CEP rules.	Ad-hoc learning algorithm
[111] Mehdiyev et al.	2015	None	Daily routine movement activities dataset	ML techniques are used to replace the manual identification of rule patterns.	Rule-based data mining methods: One-R, PIPPER, PART, Ridor and DTNB.
[118] Mousheimish et al.	2017	None	Wafer, ECG and Robots datasets	Data mining based method is proposed to learn predictive CEP rules automatically from multivariate time series.	Data mining method: Shapelets algorithm
[122] Lee et al.	2017	None	Stock trade system (NASDAQ)	SCARG framework is proposed to automatically create rules. Complex sequence events are collected and then clustered. Each cluster is graphically modelled by probabilistic model.	KNN Markov model
[123] Roldan et al.	2020	Cyber	UDP, TCP and Xmas post scans, and DoS attack	MEdit4CEP model-driven approach is used to establish data connection between IoT network and both the CEP engine and ML techniques	Linear regression, and support vector regression (SVR)
[110] Simsek et al.	2021	None	Air pollution dataset	ARECEP framework is proposed to extract CEP rules from unlabelled IoT data. (1) DL algorithms are used to label theses data as normal or anomalous. (2) The anomalous data are transformed into rules by using rule-based mining approaches.	Rule-based methods: DT, PART, ONE-R, JRIP, RIDOR, NNge or FURIA. ML/DL methods: autoencoder, CNN, RNN, LSTM, CNN-LSTM or GRU
[125] Xi et al.	2021	Physical	Terrorist activities in urban environment	Counter-terrorism early warning system was designed by combining CEP with ML to provide timely response and awareness of potential threats.	Intelligence perception (smart sensors), intelligence identification (features) and intelligence inference (CEP+ML).
[126] Roldán et al.	2021	Cyber	Subscription fuzzing, disconnection wave, TCP syn scan, UDP scan, Xmas scan and Telnet connection	Framework is proposed to integrate CEP with ML, where ML is used to enable automatic generation of CEP patterns from categorized or uncategorised data for classifying attacks or detecting anomalies. Dataset extracted from network using MQTT.	PCA is used for dimensionality reduction. Threshold value is generated based on standard deviations, mean and variance explained of the components, then Siddhi CEP engine is used to generate patterns.

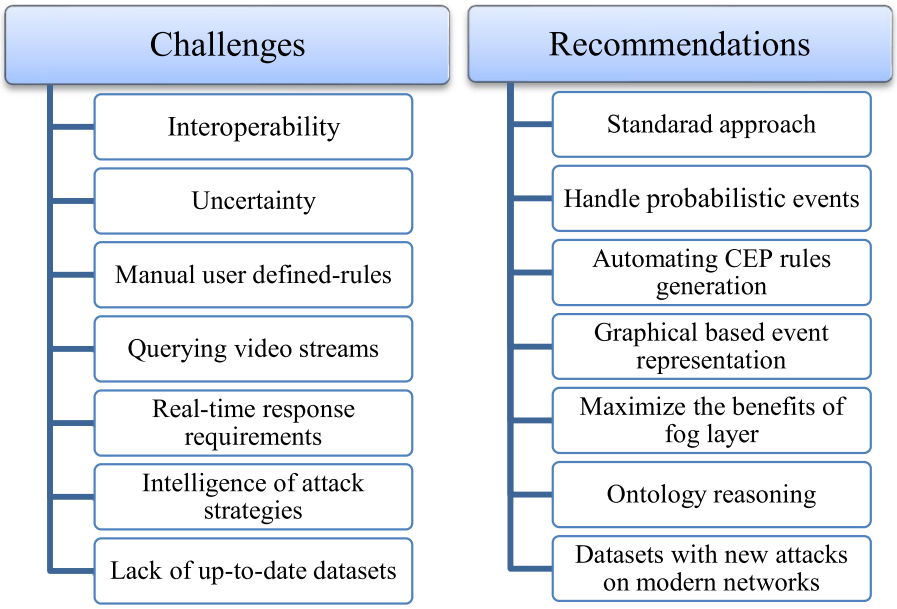


Fig. 11 The challenges experience by CEP technology for deployment of massive IoT devices with suggested recommendations

which most current methods cannot handle efficiently. Another issue is data uncertainty caused by noise, sensor error, or communication channel with possibility of receiving incomplete data streams or inaccurate data [132]. Since uncertainty is commonly treated with probabilities, a solution to this issue can be faced by extending the CEP engines to handle probabilistic events as proposed in [133–135].

- Most of the available CEP engines are based on user-defined rules, which describe the relation between observed events and events of interest. Writing such a rules maybe challenging, because it requires answering several inquiries such as which events are very relevant for detected one and which are not? Which value should they have?. The solution to such problem could be by automating the generation of the rules with the help of ML. For example in [136], historical traces of events are analysed using supervised ML techniques to derive relevant CEP rules.
- CEP has limitations when it comes to querying video streams, because of its unstructured data model and lack of an expressive query language. A possible solution to this challenge can be by using graphical based event representation of video stream to enable complex event detection with the help of deep neural networks [137], or ML [138].
- Current applications leverage CEP in cloud-based environments to provide prompt response [139], however, in near future, single application has to handle huge volume of IoT devices within short period of time. As a result of the massive amount of data generated in these future IoT applications, cloud-based solutions are incapable of meeting the low-latency real-time response requirements [139, 140]. To address this challenge, fog computing emerged as a paradigm for processing data near the network edge. For CEP systems, few research works have been focused on using fog layer as an extension for cloud-based stream processing and research gaps are still open for further investigation to maximize the benefits of fog layer [141].
- Important challenge in intrusion detection based on IoT is that as time passes, the attacks become more intelligent and diverse [142]. The current IDSs are based on pattern-based or behavior-based statistical methods, which may not provide fundamental solutions for security intrusion or attacks [142, 143]. A suggested solution by [142] is to use ontology reasoning to intelligently respond to security intrusions while employing an access control technique. In recent years, various intelligent reasoning technologies based on ontology and sematic web technology are being actively considered by researchers in intelligent systems domain for security, text mining and natural language processing [142, 144, 145].
- The current studies highlighted the lack of public up-to-date dataset that imitates new attacks on modern networks [99]. The construction of dataset is an expensive process that requires many resources and

high knowledge experts. The dataset should be constructed and updated frequently to reflect latest types of intrusions and should be made public to assist research communities.

In addition, conventional security intrusion detection and response technologies may not be effective to realize newer real-time attacks. To develop an effective IDS model, it must be trained, tested and validated against a dataset that includes both older and newer attack strategies. Developing a unified security strategy for IoT environment is a challenging task. IoT devices are often powered by batteries with limited capacity and must save energy. Some IoT devices do not have enough computing memory for executing security precautions. Securing an IoT device against certain types of attacks consumes a significant amount of energy; thus, it is critical to first identify potential threats and then implement proper countermeasures for the specific architecture of the developed IoT system [143]. ML/DL techniques such as autoencoder can be used to handle potential threats (anomaly) since it requires knowledge of normal system operation only and it will identify the threat automatically when it crosses a certain threshold value [5, 104]. CEP technology is still evolving, further research into the use of ML and DL in CEP has a huge potential. For instance, transfer learning which it is the capability to use pre-trained model for different yet similar intrusion detection can be a solution to reduce model training time for certain scenarios; however, it has not been investigated yet for physical or cyber security analysis. In addition, most of the ML/DL techniques integrated with CEP presented in literature are not implemented yet for both physical and cyber security. Another interesting research direction is the mechanism of self-healing for IoT devices, which can be used to provide suitable remediation for attack without human interventions.

Conclusion

Due to advancement of IR4.0 technologies, the dependence on datacentres to properly store, process analyse data and securely facilitate communication between different organizations increases. The deployment of IoT changes the datacentres into a more intelligent and responsive environment, however, the networked IoT devices are vulnerable to physical damage or cyber-attacks which expose datacentres to threats as well. It creates the need for more rigorous security management system such as intrusion detection. CEP technology is a promising technology to process a big data acquired by large number of IoT devices, it has the ability to process simple and complex events based on certain defined rules in quick time, and alert administrator to take preventive or corrective action. This paper provided a comprehensive review of physical and cyber intrusion detection mechanisms based on CEP technology to offer new researchers with recent progress, challenges and possible future research directions. First, the concept of CEP was discussed, and then popular CEP engines available in the market and several applications of CEP were introduced. Second, the physical and cyber threats that may experience by datacentres were highlighted. Four critical aspects that are needed to design a robust security system for datacentres against any physical or security breach, which are prevention, detection, prediction and response were discussed. ML and DL play significant roles in intrusion detection domain, the fusion of CEP with ML or DL can pave the way to produce incredible results in terms of detection for mitigating the physical and cyber security issues. This direction of research is still at its infancy and requires further investigations to find appropriate solutions. Various open issues and suggestions for solutions in CEP technology and intrusion detections have been highlighted and discussed. Future works can be done by extending the CEP engines to handle probabilistic events, automating the generation of CEP rules efficiently with the support of ML/DL, maximizing the benefits of fog computing, exploring the benefits of using transfer learning and implementing robust self-healing mechanism for IoT devices with CEP technologies. This review paper is expected to serve as a valuable resource for developing IoT based physical and cyber security system for upcoming datacentres.

Authors' contributions

Conceptualization: K. A. Alaghbari, M. H. M. Saad and M. R. Alam, Methodology: K. A. Alaghbari and M. R. Alam, Supervision: M. H. M. Saad and A. Hussain, Writing original draft: K. A. Alaghbari, Review & editing: K. A. Alaghbari, M. H. M. Saad, A. Hussain and M. R. Alam. All authors read and approved the final manuscript.

Funding

The authors acknowledge the support of the Universiti Kebangsaan Malaysia (UKM) for this research via the UKM's DIP-2020-021 research grant.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Faculty of Engineering and Technology, Multimedia University (MMU), 75450, Bukit Beruang, Melaka, Malaysia. [2]Institute of IR4.0, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia. [3]Department of Mechanical & Manufacturing Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia. [4]Department of Electrical, Electronic & System, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia. [5]Department of Computer Science and Department of Occupational Science & Occupational Therapy, University of Toronto, 27 King's College Cir, M5S 1A1 Toronto, Ontario, Canada.

Received: 10 May 2022 Accepted: 29 September 2022
Published online: 14 October 2022

References

1. Medina-Santiago A et al (2020) Adaptive model IoT for monitoring in data centers. IEEE Access 8:5622–5634. https://doi.org/10.1109/ACCESS.2019.2963061
2. Mehta G, Mittra G, Yadav VK (2018) Application of IoT to optimize data center operations. In: 2018 International Conference on Computing, Power and Communication Technologies (GUCON), pp 738–742. https://doi.org/10.1109/GUCON.2018.8674939
3. Roy A et al (2016) Energy-efficient data centers and smart temperature control system with IoT sensing. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp 1–4. https://doi.org/10.1109/IEMCON.2016.7746251
4. Elrawy M, Awad A, Hamed H (2018) Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comp 7:21. https://doi.org/10.1186/s13677-018-0123-6
5. Alaghbari KA, Saad MHM, Hussain A, Alam MR (2022) Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes. IEEE Access 10:28219–28232. https://doi.org/10.1109/ACCESS.2022.3157726
6. Amit G, Shabtai A, Elovici Y (2021) A self-healing mechanism for internet of things devices. IEEE Secur Priv 19(1):44–53. https://doi.org/10.1109/MSEC.2020.3013207
7. Singh AP, Nigam S, Gupta NK (2007) A study of next generation wireless network 6G. IntJ Innov Res Comput Commun Eng 4(1):871–874
8. Luckham D (2008) The power of events: an introduction to complex event processing in distributed enterprise systems. In: Bassiliades N, Governatori G, Paschke A (eds) Rule representation, interchange and reasoning on the web. RuleML 2008. Lecture notes in computer science, vol 5321. Springer, Berlin, Heidelberg
9. Luckham D (2011) Event processing for business: organizing the real-time enterprise. Wiley ISBN: 978-0-470-53485-4
10. Zeuch S et al (2020) Complex analytics beyond the cloud. Open J Internet Things 6(1):66–81
11. Granell C, Havlik D, Schade S, Sabeur Z, Delaney C, Pielorz J et al (2016) Future Internet technologies for environmental applications. Environ Model Softw 78:1–15. https://doi.org/10.1016/j.envsoft.2015.12.015
12. Sun AY, Zhong Z, Jeong H, Yang Q (2019) Building complex event processing capability for intelligent environmental monitoring. Environ Model Softw 116:1–6. https://doi.org/10.1016/j.envsoft.2019.02.015
13. Huang Y, Williams BC, Zheng L (2011) Reactive model-based monitoring in RFID-enabled manufacturing. Comput Ind 62(9):811–819. https://doi.org/10.1016/j.compind.2011.08.003
14. Dhillon A, Majumdar S, St-Hilaire M, El-Haraki A (2018) MCEP: a mobile device based complex event processing system for remote healthcare. In: Proc. IEEE Int. Conf. Internet Things (ICIOT), pp 203–210. https://doi.org/10.1109/Cybermatics_2018.2018.00064
15. Lan L, Shi R, Wang B, Zhang L, Jiang N (2019) A universal complex event processing mechanism based on edge computing for internet of things real-time monitoring. IEEE Access 7:101865–101878. https://doi.org/10.1109/ACCESS.2019.2930313
16. Saad MHM, Sarker MR, Hussain A (2020) Application of complex event processing approaches for intelligent building development: a review. J Ambient Intell Smart Environ 12(2):101–124. https://doi.org/10.3233/AIS-200555
17. Tawsif K, Hossen J, Raja JE, Jesmeen MZH, Arif EMH (2018) A review on complex event processing systems for big data. In: 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), pp 1–6. https://doi.org/10.1109/INFRKM.2018.8464787
18. Wanner J, Wissuchek C, Janiesch C (2019) Machine learning and complex event processing. A review of real-time data analytics for the industrial internet of things. Enterp Model Inf Syst Arch 15:1. Berlin: Gesellschaft für Informatik e.V.. (S. 1-27). https://doi.org/10.18417/emisa.15.19/INFRKM.2018.8464787
19. Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M (2020) A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Commun Surv Tutor 22(3):1646–1685. https://doi.org/10.1109/COMST.2020.2988293
20. Mishra N, Pandya S (2021) Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review. IEEE Access 9:59353–59377. https://doi.org/10.1109/ACCESS.2021.3073408
21. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surv Tutor 21(3):2671–2701. https://doi.org/10.1109/COMST.2019.2896380
22. Khraisat A, Gondal I, Vamplew P et al (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2:20. https://doi.org/10.1186/s42400-019-0038-7
23. Arora S, Gambheer R, Vohra M (2021) Design of secure IoT systems: a practical approach across industries. McGraw-Hill education, USA, ISBN: 9781260463101
24. Cowan C, Gaskins C (2006) Monitoring physical threats in the data center, APC white paper
25. Sethi P, Sarangi SR (2017) Internet of things: architectures, protocols, and applications. J Electr Comput Eng 2017:9324035, 25 pages. https://doi.org/10.1155/2017/9324035
26. Mashal I, Alsaryrah O, Chung T-Y, Yang C-Z, Kuo W-H, Agrawal DP (2015) Choices for interaction with things on Internet and underlying issues. Ad Hoc Netw 28:68–90. https://doi.org/10.1016/j.adhoc.2014.12.006
27. Wu M, Lu T-J, Ling F-Y, Sun J, Du H-Y (2010) Research on the architecture of internet of things. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE'10), vol 5. IEEE, Chengdu, pp V5-484–V5-487. https://doi.org/10.1109/ICACTE.2010.5579493
28. Al-Fuqaha A, Guizani M, Mohammadi M et al (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor 17(4):2347–2376. https://doi.org/10.1109/COMST.2015.2444095
29. Khan R, Khan SU, Zaheer R et al (2012) Future internet: the internet of things architecture, possible applications and key challenges. In: IEEE 10th international conference on frontiers of information technology, pp 257–260. https://doi.org/10.1109/FIT.2012.53
30. Iqbal MA, Hussain S, Xing H, Imran MA (2020) Enabling the internet of things: fundamentals, design and applications, 1st edn. IEEE press, Wiley, Hoboken, ISBN: 978-1-119-70125-5
31. Kumar NM, Dash A, Singh NK (2018) Internet of things (IoT): an opportunity for energy-food-water nexus. In: IEEE International Conference on Power Energy, Environment and Intelligent Control (PEEIC), pp 68–72. https://doi.org/10.1109/PEEIC.2018.8665632
32. Kumar NM, Mallick PK (2018) The internet of things: insights into the building blocks, component interactions, and architecture layers. Procedia Comput Sci 132:109–117. https://doi.org/10.1016/j.procs.2018.05.170
33. Sarker IH, Khan AI, Abushark YB et al (2022) Internet of things (IoT) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mobile Netw Appl. https://doi.org/10.1007/s11036-022-01937-3
34. Wang Y, Cao K (2012) Context-aware complex event processing for event cloud in internet of things. In: 2012 International Conference on Wireless Communications and Signal Processing (WCSP), pp 1–6. https://doi.org/10.1109/WCSP.2012.6542861
35. Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019) A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 7:82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045
36. Chatterjee J, Das A, Ghosh S, Das MK, Bag R (2020) Chapter 8: a review of cyber attack analysis and security aspect of IoT-enabled technologies. In: IoT: security and privacy paradigm. CRC Press, Taylor & Francis Group, p 159, Boca Raton, FL, United States
37. Siboni S, Cohen A (2020) Anomaly detection for individual sequences with applications in identifying malicious tools. Entropy 22:649. https://doi.org/10.3390/e22060649
38. Silva SS, Silva RM, Pinto RC, Salles RM (2013) Botnets: a survey. Comput Netw 57(2):378–403. https://doi.org/10.1016/j.comnet.2012.07.021
39. Abbasi FH, Harris RJ, Moretti G, Haider A, Anwar N (2012) Classification of malicious network streams using honeynets. In: Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, pp 891–897. https://doi.org/10.1109/GLOCOM.2012.6503226

40. Akbar A, Khan A, Carrez F, Moessner K (2017) Predictive analytics for complex IoT data streams. IEEE Internet Things J 4(5):1571–1582. https://doi.org/10.1109/JIOT.2017.2712672
41. Cugola G, Margara A (2012) Processing flows of information: from data stream to complex event processing. ACM Comput Surv 44:1–62. https://doi.org/10.1145/2187671.2187677
42. Esper, https://www.espertech.com/esper. Accessed 20 Sept 2022
43. Siddhi CEP engine, https://siddhi.io/en/v5.1/docs/. Accessed 29 July 2022
44. Saad MHM (2017) Pemprosesan Peristiwa Kompleks Untuk Aplikasi Sistem Kejuruteraan Pintar, PhD. Thesis, Universiti Kebangsaan Malaysia, Malaysia
45. Shahad RA, Saad MHM, Hussain A (2018) Activity recognition for smart building application using complex event processing approach. Int J Adv Sci Eng Inf Technol 8(2). https://doi.org/10.18517/ijaseit.8.2.2575
46. Wongsuphasawat K, Plaisant C, Taieb-Maimon M, Shneiderman B (2012) Querying event sequences by exact match or similarity search: design and empirical evaluation. Interact Comput 24(2):55–68. https://doi.org/10.1016/j.intcom.2012.01.003
47. Merigo MJ, Gil-lafuente AM (2012) Decision-making techniques with similarity measures and OWA operators. Stat Oper Res Trans 36(1):81–102 https://raco.cat/index.php/SORT/article/view/254885
48. Moen P (2000) Attribute, event sequence and event type similarity notions for data mining. PhD thesis, Dept. of Computer Science, University of Helsinki, Finland
49. Mei Y, Madden S (2009) ZStream: a cost-based query processor for adaptively detecting composite events categories and subject descriptors. In: Proc. 35th SIGMOD Int. Conf. Manag. data, pp 193–206. https://doi.org/10.1145/3448016.3457245
50. Agrawal R, Lin K, Sawhney HS, Shim K (1995) Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In: Proc. 21st Int. Conf. Very Large Databases, pp 490–501
51. Pooja KS, Chandrashekar KT, Thungamani M, Gireesh Babu CN, Is AW, Home AS (2015) Complex event processing in smart homes, no. 3, pp 544–550 ISSN: 2395-3470
52. Xu M, Liu Z, Li J (2014) Tree-structured network based hierarchical complex event processing in wireless sensor networks. In: 2014 Asia-Pacific services computing conference, pp 185–190. https://doi.org/10.1109/APSCC.2014.38
53. Xiao F, Zhan C, Lai H, Tao L, Qu Z (2017) New parallel processing strategies in complex event processing systems with data streams. Int J Distrib Sens Netw. https://doi.org/10.1177/1550147717728626
54. Saleh O (2013) Complex event processing in wireless sensor networks. In: 25th GI-workshop on foundations of databases, pp 69–74
55. Lima M, Lima R, Lins F, Bonfim M (2022) Beholder – A CEP-based intrusion detection and prevention systems for IoT environments. Comput Secur 120:102824. https://doi.org/10.1016/j.cose.2022.102824
56. Jun C, Chi C (2014) Design of complex event-processing IDS in internet of things. In: 2014 sixth international conference on measuring technology and mechatronics automation, pp 226–229. https://doi.org/10.1109/ICMTMA.2014.57
57. Marques da Silva Cardoso A, Fernandes Lopes R, Soares Teles A, Benedito Veras Magalhães F (2018) Real-time DDoS detection based on complex event processing for IoT. In: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), pp 273–274. https://doi.org/10.1109/IoTDI.2018.00036
58. Raj R, Sahu RK, Chaudhary B, Prasad BR, Agarwal S (2017) Real time complex event processing and analytics for smart building. In: 2017 Conference on Information and Communication Technology (CICT), pp 1–6. https://doi.org/10.1109/INFOCOMTECH.2017.8340593
59. Alseiari FAA, Aung Z (2015) Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining. In: 2015 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE). IEEE, Offenburg, pp 148–153. https://doi.org/10.1109/ICSGCE.2015.7454287
60. Alaghbari KA, Hanif Md Saad M, Hussain A, Othman RA, Alam MR (2021) A comparison of sequential prediction algorithms in IoT enabled smart environments. In: 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), pp 238–243. https://doi.org/10.1109/ICSGRC53186.2021.9515261
61. Shahad RA, Ibrahim MF, Xian EL, Hussain A, Saad MHM (2018) Suspicious loitering detection from annotated CCTV feed using CEP based approach. Jurnal Kejuruteraan 30(1):83–91. https://doi.org/10.17576/jkukm-2018-30(1)
62. Itria ML, Daidone A, Ceccarelli A (2014) A Complex event processing approach for crisis management systems, computer science, ArXiv preprint. https://doi.org/10.48550/arXiv.1404.7551
63. Complex event example, https://developer.hpe.com/blog/better-complex-event-processing-at-scale-using-a-microservices-based-str/. Accessed 17 Sept 2021
64. LaPiedra J (2002) The Information Security Process Prevention, Detection and Response, global information assurance certification paper, SANS institute, Maryland, United States
65. Modern datacentre example, https://www.black-box.eu/en-int/page/43877/Resources/technical/Black-Box-Explains/lan/What-is-Server-Room-Environmental-Monitoring. Accessed 14 Mar 2022
66. Yamanoue T (2020) Monitoring of servers and server rooms by IoT system that can configure and control its terminal sensors behind a NAT using a Wiki Page on the internet. J Inf Process 28:204–213. https://doi.org/10.2197/ipsjjip.28.204
67. PukiWiki, https://en.wikipedia.org/wiki/PukiWiki. Accessed 7 Sept 2021
68. Yamanoue T, Oda K, Shimozono K (2013) A malicious bot capturing system using a beneficial bot and Wiki. J Inf Process 21(2):237–245. https://doi.org/10.2197/ipsjjip.21.237
69. Kaya M, Kaya YC (2017) Complex event processing using IoT devices based on Arduino. Int J Cloud Comput Serv Arch 7(6). https://doi.org/10.5121/ijccsa.2017.7602
70. Amru SR, Saad MHM, Kamal N, Hussain A (2018) Real time event detection for intelligent building surveillance system application. In: Proceedings of undergraduate research 2018, Bangi, Selangor, Malaysia. https://doi.org/10.5281/zenodo.2528454
71. Mijović V, Tomašević N, Janev V et al (2019) Emergency management in critical infrastructures: a complex-event-processing paradigm. J Syst Sci Syst Eng 28:37–62. https://doi.org/10.1007/s11518-018-5393-5
72. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15. https://doi.org/10.1145/1541880.1541882
73. Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 16(1):303–336. https://doi.org/10.1109/SURV.2013.052213.00046
74. Lakshmi KN, Neema N, Mohammed Muddasir N, Prashanth MV (2020) Anomaly detection techniques in data mining—a review. In: Ranganathan G, Chen J, Rocha Á (eds) Inventive communication and computational technologies. Lecture notes in networks and systems, vol 89. Springer, Singapore. https://doi.org/10.1007/978-981-15-0146-3_76
75. Hong J, Liu C, Govindarasu M (2014) Integrated anomaly detection for cyber security of the substations. IEEE Trans Smart Grid 5(4):1643–1653. https://doi.org/10.1109/TSG.2013.2294473
76. Mishra P, Pilli ES, Varadharajan V, Tupakula U (2017) Intrusion detection techniques in cloud environment: a survey. J Netw Comput Appl 77:18–47. https://doi.org/10.1016/j.jnca.2016.10.015
77. Han J, Kamber M, Pei J (eds) (2012) Data mining: concepts and techniques. Morgan Kaufmann, Boston
78. Duque S, bin Omar MN (2015) Using data mining algorithms for developing a model for intrusion detection system (IDS). Procedia Comput Sci 61:46–51. https://doi.org/10.1016/j.procs.2015.09.145
79. Feng W, Zhang Q, Hu G, Huang JX (2014) Mining network data for intrusion detection through combining SVMs with ant colony networks. Futur Gener Comput Syst 37:127–140. https://doi.org/10.1016/j.future.2013.06.027
80. Yu PS, Tsia JJ (2009) Machine learning in cyber trust: security, privacy, and reliability, 1st edn. Springer US, Springer-Verlag US, pp 1–362. https://doi.org/10.1007/978-0-387-88735-7
81. Nishani L, Biba M (2016) Machine learning for intrusion detection in MANET: a state-of-the-art survey. J Intell Inf Syst 46(2):391–407. https://doi.org/10.1007/s10844-015-0387-y
82. Namdev N, Agrawal S, Silkari S (2015) Recent advancement in machine learning based internet traffic classification. Procedia Comput Sci 60:784–791. https://doi.org/10.1016/j.procs.2015.08.238
83. Tan P-N, Steinbach M, Karpatne A, Kumar V (2018) Introduction to data mining, 2nd edn. Pearson, Hudson Street, NY, USA

84. Amin SO, Siddiqui MS, Hong CS, Lee S (2009) RIDES: robust intrusion detection system for ip-based ubiquitous sensor networks. Sensors 9(5):3447. https://doi.org/10.3390/s90503447
85. Muzammil MJ, Qazi S, Ali T (2013) Comparative analysis of classification algorithms performance for statistical based intrusion detection system. In: 2013 3rd IEEE International Conference on Computer, Control and Communication (IC4), Karachi, pp 1–6. https://doi.org/10.1109/IC4.2013.6653738
86. Mabu S, Chen C, Lu N, Shimada K, Hirasawa K (2011) An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans Syst Man Cybern Part C Appl Rev 41(1):130–139. https://doi.org/10.1109/TSMCC.2010.2050685
87. Moshtaghi M, Bezdek JC, Leckie C, Karunasekera S, Palaniswami M (2015) Evolving fuzzy rules for anomaly detection in data streams. IEEE Trans Fuzzy Syst 23(3):688–700. https://doi.org/10.1109/TFUZZ.2014.2322385
88. Aniello L, Lodi G, Baldoni R (2011) Inter-domain stealthy port scan detection through complex event processing. In: Proceedings of the 13th European Workshop on Dependable Computing EWDC '11, ACM New York, pp 67–72. https://doi.org/10.1145/1978582.1978597
89. Cheng S, Cheng Z, Luan Z, Qian D (2011) NEPnet: a scalable monitoring system for anomaly detection of network service. In: 7th International Conference on Network and Service Management (CNSM)
90. Gad R, Kappes M, Boubeta-Puig J, Medina-Bulo I (2013) Employing the CEP paradigm for network analysis and surveillance. In: Proceedings of the ninth advanced international conference on telecommunications. IARIA, Rome, pp 204–210
91. Jayan K, Rajan AK (2014) Preprocessor for complex event processing system in network security. In: 2014 fourth international conference on advances in computing and communications, pp 187–189. https://doi.org/10.1109/ICACC.2014.52
92. Mohan R, Vaidehi V, Ajay Krishna A, Mahalakshmi M, Chakkaravarthy SS (2015) Complex event processing based hybrid intrusion detection system. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp 1–6. https://doi.org/10.1109/ICSCN.2015.7219827
93. Vegh L, Miclea L (2016) Complex event processing for attack detection in a cyber-physical system. In: 2016 IEEE international conference on automation, quality and testing, robotics (AQTR), pp 1–6. https://doi.org/10.1109/AQTR.2016.7501296
94. Devi BSK, Subbulakshmi T (2021) Cloud DDoS detection and defense system using complex event processing. In: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp 118–128. https://doi.org/10.1109/ICICCS51141.2021.9432102
95. Taylor H, Yochem A, Phillips L, Martinez F (2009) Event-driven architecture: how SOA enables the real time enterprise. Addison-Wesley Professional, Boston ISBN: 9780321591388
96. Prasad R, Rohokale V (2020) Artificial intelligence and machine learning in cyber security. In: Cyber security: the lifeline of information and communication technology. Springer, New York, pp 231–247. https://doi.org/10.1007/978-3-030-31703-4_16
97. Lew J, Shah DA, Pati S et al (2019) Analyzing machine learning workloads using a detailed GPU simulator. In: Paper presented at: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, Madison, pp 151–152. https://doi.org/10.1109/ISPASS.2019.00028
98. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. J Big Data 2(1):1. https://doi.org/10.1186/s40537-014-0007-7
99. Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F (2021) Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech 32(1):e4150. https://doi.org/10.1002/ett.4150
100. Shareena J, Ramdas A, Haripriya AP et al (2021) Intrusion detection system for IOT botnet attacks using deep learning. SN Comput Sci 2:205. https://doi.org/10.1007/s42979-021-00516-9
101. Anthi E, Williams L, Słowińska M, Theodorakopoulos G, Burnap P (2019) A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J 6(5):9042–9053. https://doi.org/10.1109/JIOT.2019.2926365
102. Meidan Y, Bohadana M, Shabtai A, Ochoa M, Tippenhauer NO, Guarnizo JD, Elovici Y (2017), arXiv preprint) Detection of unauthorized IoT devices using machine learning techniques. https://doi.org/10.48550/arXiv.1709.04647
103. Doshi R, Apthorpe N, Feamster N (2018) Machine learning DDoS detection for consumer internet of things devices. In: IEEE security and privacy workshops (SPW). IEEE, pp 29–35. https://doi.org/10.1109/SPW.2018.00013
104. Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y (2018) N-BaIoT network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput 17(3):12–22. https://doi.org/10.1109/MPRV.2018.03367731
105. Yuan X, Li C, Li X (2017) Deep defense: identifying DDoS attack via deep learning. In: 2017 IEEE international conference on smart computing (SMARTCOMP), Hong Kong, pp 1–8. https://doi.org/10.1109/SMARTCOMP.2017.7946998
106. Ibitoye O, Shafiq O, Matrawy A (2019) Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In: IEEE global communications conference (GLOBECOM), 2019, pp 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9014337
107. Thamilarasu G, Chawla S (2019) Towards deep-learning-driven intrusion detection for the internet of things. Sensors 19:1977. https://doi.org/10.3390/s19091977
108. Alakari A, Li KF, Gebali F (2020) A situation refinement model for complex event processing. Knowl-Based Syst 198:105881. https://doi.org/10.1016/j.knosys.2020.105881
109. Margara A, Cugola G, Tamburrelli G (2014) Learning from the past: automated rule generation for complex event processing. In: Proceedings of the 8th ACM international conference on distributed event-based systems, pp 47–58. https://doi.org/10.1145/2611286.2611289
110. Simsek MU, Yildirim F, Ozdemir S (2021) A deep learning-based CEP rule extraction framework for IoT data. J Supercomput 8:8563–8592. https://doi.org/10.1007/s11227-020-03603-5
111. Mehdiyev N, Krumeich J, Enke D, Werth D, Loos P (2015) Determination of rule patterns in complex event processing using machine learning techniques. Procedia Comput Sci 61:395–401. https://doi.org/10.1016/j.procs.2015.09.168
112. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11(1):63–90. https://doi.org/10.1023/A:1022631118932
113. Cohen WW (1995) Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning. https://doi.org/10.1016/B978-1-55860-377-6.50023-2
114. Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, pp 144–151
115. Martin B (1995) Instance-based learning: nearest neighbour with generalisation. Working paper series. University of Waikato Hamilton, New Zealand https://hdl.handle.net/10289/1095
116. Gaines BR, Compton P (1995) Induction of ripple-down rules applied to modeling large databases. J Intell Inf Syst 5(3):211–228. https://doi.org/10.1007/BF00962234
117. Hall M, Frank E (2008) Combining naive bayes and decision tables. In: Wilson DL, Chad H (eds) Proceedings of Twenty-First International Florida Artificial Intelligence Research Society Conference. AAAI Press, Coconut Grove, pp 318–319 https://hdl.handle.net/10289/1773
118. Mousheimish R, Taher Y, Zeitouni K (2017) Automatic learning of predictive CEP rules: bridging the gap between data mining and complex event processing. In: Proceedings of the 11th ACM international conference on distributed and event-based systems, pp 158–169. https://doi.org/10.1145/3093742.3093917
119. Ye L, Keogh E (2009) Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 947–956. https://doi.org/10.1145/1557019.1557122
120. Olszewski RT (2001) Generalized feature extraction for structural pattern recognition in time-series data. Technical report. DTIC Document, Carnegie Mellon University ISBN: 978-0-493-53871-6
121. Bache K, Lichman M (2013) UCI machine learning repository. University of California, Irvine http://archive.ics.uci.edu/ml

122. Lee OJ, Jung JE (2017) Sequence clustering-based automated rule generation for adaptive complex event processing. Futur Gener Comput Syst 66:100–109. https://doi.org/10.1016/j.future.2016.02.011
123. Roldán J, Boubeta-Puig J, Martínez JL, Ortiz G (2020) Integrating complex event processing and machine learning: an intelligent architecture for detecting IoT security attacks. Expert Syst Appl 149:113251. https://doi.org/10.1016/j.eswa.2020.113251
124. Boubeta-Puig J, Ortiz G, Medina-Bulo I (2015) MEdit4CEP: a model-driven solution for real-time decision making in SOA 2.0. Knowl-Based Syst 89:97–112. https://doi.org/10.1016/j.knosys.2015.06.021
125. Xi M, Lingyu N, Jiapeng S (2021) Research on urban anti-terrorism intelligence perception system from the perspective of Internet of things application. Int J Electr Eng Educ 58(2):248–257. https://doi.org/10.1177/0020720918819247
126. Roldán-Gómez J, Boubeta-Puig J, Castelo Gómez JM, Carrillo-Mondéjar J, Martínez Martínez JL (2021) Attack pattern recognition in the internet of things using complex event processing and machine learning. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 1919–1926. https://doi.org/10.1109/SMC52423.2021.9658711
127. Roldán-Gómez J, Boubeta-Puig J, Pachacama-Castillo G, Ortiz G, Martínez JL (2021) Detecting security attacks in cyber-physical systems: a comparison of Mule and WSO2 intelligent IoT architectures. PeerJ Comput Sci 7:e787, 1–35. https://doi.org/10.7717/peerj-cs.787
128. Milosevic Z, Chen W, Berry A, Rabhi FA (2016) An open architecture for event-based analytics. Int J Data Sci Anal 2:13–27. https://doi.org/10.1007/s41060-016-0029-7
129. R: the R project for statistical computing. http://www.r-project.org/. Accessed 1 Aug 2022
130. Azure. https://azure.microsoft.com/en-us/documentation/articles/stream-analytics-introduction/. Accessed 1 Aug 2022
131. Fielding RT (2000) Architectural styles and the design of network-based software architectures, PhD thesis. University of California, Irvine
132. Wang YH, Cao K, Zhang XM (2013) Complex event processing over distributed probabilistic event streams. Comput Math Appl 66(10):1808–1821. https://doi.org/10.1016/j.camwa.2013.06.032
133. Shen Z, Kawashima H, Kitagawa H (2008) Probabilistic event stream processing with lineage. In: Proceedings of the data engineering workshop. https://doi.org/10.1109/MDMW.2008.12
134. Chuanfei X, Shukuan L, Lei W, Jianzhong Q (2010) Complex event detection in probabilistic stream. In: 2010 12th International Asia-Pacific Web Conference, pp 361–363. https://doi.org/10.1109/APWeb.2010.56
135. Kawashima H, Kitagawa H, Li X (2010) Complex event processing over uncertain data streams. In: Proceedings of the fifth international conference on P2P, parallel, grid, cloud and internet computing, pp 521–526. https://doi.org/10.1109/3PGCIC.2010.89
136. Margara A, Cugola G, Tamburrelli G, Lugano I (2013) Towards automated rule learning for complex event processing. Technical report. VU University, Amsterdam
137. Yadav P, Curry E (2019) VidCEP: complex event processing framework to detect spatiotemporal patterns in video streams. In: 2019 IEEE international conference on big data (big data), pp 2513–2522. https://doi.org/10.1109/BigData47090.2019.9006018
138. Li Z, Katsifodimos A, Bozzon A, Houben GJ (2020) Complex event processing on real-time video streams. In: CEUR workshop proceedings, p 2652 Virtual, online, Japan
139. Higashino WA, Capretz MAM, Bittencourt LF (2016) CEPSim: modelling and simulation of complex event processing systems in cloud environments. Futur Gener Comput Syst 65:122–139. https://doi.org/10.1016/j.future.2015.10.023
140. Ziehn A (2020) Complex event processing for the internet of things. In: Proceedings of the VLDB 2020 PhD Workshop, German Research Centre for Artificial Intelligence (DFKI), Germany. https://www.dfki.de/fileadmin/user_upload/import/11130_PhD_Workshop.pdf
141. Mondragón-Ruiz G, Tenorio-Trigoso A, Castillo-Cara M et al (2021) An experimental study of fog and cloud computing in CEP-based real-time IoT applications. J Cloud Comp 10:32. https://doi.org/10.1186/s13677-021-00245-7
142. Choi C, Choi J (2019) Ontology-based security context reasoning for power IoT-cloud security service. IEEE Access 7:110510–110517. https://doi.org/10.1109/ACCESS.2019.2933859
143. Balogh S, Gallo O, Ploszek R, Špaček P, Zajac P (2021) IoT security challenges: cloud and blockchain, postquantum cryptography, and evolutionary techniques. Electronics 10:2647. https://doi.org/10.3390/electronics10212647
144. Zheng H, Wang Y, Han C, Le F, He R, Lu J (2018) Learning and applying ontology for machine learning in cyber attack detection. In: 2018 17th IEEE Int. Conf. On Trust, Security And Privacy In Comp. And Comm./12th IEEE Int. Conf, On Big Data Sci. and Eng. (TrustCom/BigDataSE), pp 1309–1315. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00180
145. Hohenecker P, Lukasiewicz T (2020) Ontology reasoning with deep neural networks. J Artif Intell Res 68:503–540. https://doi.org/10.1613/jair.1.11661

Publisher's Note