Smart Sensors, Measurement and Instrumentation 43

Shantanu Pal
Zahra Jadidi
Ernest Foo *Editors*

# Secure and Trusted Cyber Physical Systems

## Recent Approaches and Future Directions

Springer

# Smart Sensors, Measurement and Instrumentation

Volume 43

**Series Editor**

The Smart Sensors, Measurement and Instrumentation series (SSMI) publishes new developments and advancements in the fields of Sensors, Instrumentation and Measurement technologies. The series focuses on all aspects of design, development, implementation, operation and applications of intelligent and smart sensors, sensor network, instrumentation and measurement methodologies. The intent is to cover all the technical contents, applications, and multidisciplinary aspects of the field, embedded in the areas of Electrical and Electronic Engineering, Robotics, Control, Mechatronics, Mechanical Engineering, Computer Science, and Life Sciences, as well as the methodologies behind them. Within the scope of the series are monographs, lecture notes, selected contributions from specialized conferences and workshops, special contribution from international experts, as well as selected PhD theses.

Indexed by SCOPUS and Google Scholar.

Shantanu Pal · Zahra Jadidi · Ernest Foo
Editors

# Secure and Trusted Cyber Physical Systems

## Recent Approaches and Future Directions

Springer

*Editors*

Shantanu Pal
School of Computer Science
Queensland University of Technology
Brisbane, QLD, Australia

Ernest Foo
School of Information and Communication Technology
Griffith University
Gold Coast, QLD, Australia

Zahra Jadidi
School of Information and Communication Technology
Griffith University
Gold Coast, QLD, Australia

ISSN 2194-8402 ISSN 2194-8410 (electronic)
Smart Sensors, Measurement and Instrumentation
ISBN 978-3-031-08269-6 ISBN 978-3-031-08270-2 (eBook)
https://doi.org/10.1007/978-3-031-08270-2

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Cyber Physical Systems (CPSs) have changed the way people interact with industrial systems. CPS deploys sensing, computation, and control and networking technology required for the integration of CPS. With the development of the Internet of Things (IoT) applications and services, there is a tremendous demand for these services and applications in the fourth industrial revolution (Industry 4.0). IoT-enabled CPS brings high benefits of connecting processes and smart devices to gather and share data. The cybersecurity of IoT-enabled CPS is a critical challenge, as the increasing number of Internet-connected devices in CPS creates a broader attack surface. Therefore, sensing data from the physical environment, securing the sensed data, and delivering it to the authorized users are significant in these environments.

CPS is composed of heterogeneous devices, e.g., sensors, actuators, and embedded systems. There are also various types of software and firmware to control and monitor CPS networks. Connection to corporate networks and the public Internet creates multiple security, privacy, and trust issues, and it exposes CPS to big, complex, and distributed networks that are vulnerable to new threats. In these environments, vulnerabilities can be in the CPS components themselves or the communication protocols they use. Advanced attacks on CPSs can affect both the cyber and physical domains. To understand the security level of a CPS network, it is crucial to understand the security vulnerabilities of hardware/software components, potential threats, and available defence methods. This information can be used to design and implement an efficient and reliable security architecture. Therefore, this book aims to provide an overview of various security challenges in CPS and discuss the possible solutions that could mitigate those challenges.

The complete book is composed of seven chapters. These chapters contain a wide range of information from communication technologies, policies for information security control, defence mechanisms against security threats and attacks and novel applications related to different domains in CPS.

Chapter "Challenges and Opportunities of Blockchain for Cyber Threat Intelligence Sharing" investigates how blockchain technology can address many challenges in the existing Cyber Threat Intelligence (CTI) sharing platforms. The chapter presents several general CTI sharing challenges to determine the role of blockchain-based sharing moving forward. It discusses how blockchain can bring opportunities to address these challenges securely and efficiently. The chapter also highlights relevant works in this domain and outlines some unique future research questions.

Chapter "System Identification Methods for Industrial Control Systems" presents a systematic discussion of the recent development in system identification from the automatic control perspective. In the beginning, the chapter presents a classification of design features of Industrial Control Systems (ICSs). The classification of ICSs allows identifying limitations and new challenges in the literature on system identification techniques. Then the chapter reviews the literature on system identification techniques for creating models of ICSs.

Chapter "Vulnerability Management in IIoT-Based Systems: What, Why and How" presents the correlation between the IIoT and Supervisory Control and Data Acquisition (SCADA) systems, followed by security challenges faced by IIoT-based systems. The chapter emphasizes the role of Vulnerability Management (VM) in securing the critical systems, followed by studying the state of art approaches for VM. The chapter underscores the design challenges and research opportunities for efficiently managing the increasing vulnerabilities. Finally, the chapter discusses the future research directions for developing techniques for efficient VM.

Chapter "Review of Cyber Security for Power Trading and Communication Systems" provides a systematic review of deployments of security mechanisms for energy market trading and communication systems. This review is categorized into four themes: (1) security technologies that can be applied to energy trading and call audit systems, (2) blockchain technology that can be applied to protect energy trading and auditing services, (3) communication technology (voice over IP and video conferencing) that operates in the cloud, and (4) network performance and security management for voice over IP and video conferencing systems. In addition, this study explores the use of blockchain technology that has increasingly emerged in a microgrid (peer-to-peer) energy trading and reveals a gap in using blockchain for microgrid national energy trading. Finally, this study emphasizes balancing network security and performance when systems are hosted in the cloud.

Chapter "DDoS Threats and Solutions for 5G-Enabled IoT Networks" presents the significant security challenges for 5G-Enabled IoT Networks. For example, the seamless connectivity of 5G could be a security threat allowing attacks, e.g., distributed denial of service (DDoS), because attackers might have easy access to the network infrastructure and higher bandwidth to enhance the effects of the attack. This chapter studies the DDoS attacks and classification of DDoS in detail. It also discusses some general approaches proposed to mitigate DDoS threats. Finally, this chapter covers strategies using SDN in 5G enabled IoT network platforms.

Chapter "A Lightweight Blockchain-Based Trust Management Framework for Access Control in IoT" proposes a lightweight blockchain-based trust management framework for IoT devices. The framework is built upon high resource devices to form the underlying Peer-to-Peer (P2P) network. In addition, a smart contract mechanism to generate a trustworthy environment for IoT devices is developed. Finally, the authors proposed a reputation-based consensus algorithm with the trust evaluation approach that can significantly decrease the mining time. Simulations have demonstrated that the proposed framework achieves low delay time, high Transactions Per Second (TPS), and less processing time than relevant baselines. Furthermore, the work shows that the proposed framework is resilient to several security attacks in blockchain systems.

Chapter "Utilising K-Means Clustering and Naive Bayes for IoT Anomaly Detection: A Hybrid Approach" discusses the significance of IoT anomaly detection. It suggests a potential alternative anomaly detection algorithm to be implemented within IoT systems that can be applied across different types of devices. The proposed algorithm comprises both unsupervised and supervised areas of machine learning, utilizing the most substantial facets of each methodology. The chapter also presents a detailed experimental result of the proposed algorithm's effectiveness to classify attacks.

We would like to express our sincere gratitude toward the Queensland University of Technology and Griffith University for providing the support that made this book possible. We would also like to thank all the researchers who contributed to the chapters of this book.

Brisbane, Australia

Dr. Shantanu Pal
Dr. Zahra Jadidi
Dr. Ernest Foo

# Contents

# Challenges and Opportunities of Blockchain for Cyber Threat Intelligence Sharing

**Kealan Dunnett, Shantanu Pal, and Zahra Jadidi**

**Abstract** The emergence of the Internet of Things (IoT) technology has caused a powerful transition in the cyber threat landscape. As a result, organisations have had to find new ways to better manage the risks associated with their infrastructure. In response, a significant amount of research has focused on developing efficient Cyber Threat Intelligence (CTI) sharing platforms. However, most existing solutions are highly centralised and do not provide a way to exchange information in a distributed way. In this chapter, we subsequently seek to evaluate how blockchain technology can be used to address a number of limitations present in existing CTI sharing platforms. To determine the role of blockchain-based sharing moving forward, we present a number of general CTI sharing challenges, and discuss how blockchain can bring opportunities to address these challenges in a secure and efficient manner. Finally, we discuss a list of relevant works and note some unique future research questions.

**Keywords** Blockchain · Cyber threat intelligence · Security · Information sharing

## 1 Introduction

Each year the threat landscape continues to evolve in both the types of cyber-attacks and the methods used to commit them [18]. Organisations have subsequently had to find ways to manage the increased risk associated with the infrastructure they depend

K. Dunnett (✉) · S. Pal
School of Computer Science, Queensland University of Technology, Brisbane, QLD 4000, Australia
e-mail: kealan.dunnett@connect.qut.edu.au

S. Pal
e-mail: shantanu.pal@qut.edu.au

Z. Jadidi
School of Information and Communication Technology, Griffith University, Gold Coast Campus, QLD 4222, Australia
e-mail: z.jadidi@griffith.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_1

on to operate. As a result, several Cyber Security Risk Management (CSRM) frameworks have been developed to define a more concrete framework to better manage this risk [17]. However, with the emergence of the Internet of Things (IoT) technology, smart portable sensors and their resource-constrained nature, the threat landscape has recently grown at a rate that makes the traditional CSRM task challenging [29, 34]. To minimise cyber threats, organisations continue to develop methods focused on gathering threat-based information specific to them. Towards this, Cyber Threat Intelligence (CTI) is a concept that describes the collection and analysis of threat information by an organisation. The emergence of CTI in recent years has seen its integration into traditional CSRM frameworks become a effective threat mitigation strategy [16].

The SANS institute is a US based organisation that conducts a yearly CTI survey across industry. The primary aim of this survey is to understand the current state of CTI use within industry. In their 2021 survey, a significant milestone was reported, 100% of surveyed organisations indicated that they either currently do or have plans to use CTI in some way [6]. When this figure is contrasted with the 75% reported only four year earlier in 2017, it is clear that CTI will continue to play a critical role in threat mitigation within industry moving forward.

Sharing CTI cooperatively between organisations can be highlighted as a mutually beneficial process for all participating organisations [15]. However, in practise CTI sharing is challenging due to the variety of ways threats can affect the components that make up an organisations infrastructure (e.g., Storage, Networks and Communication). For example, the man-in-the-middle attack, eavesdropping attack, phishing and spear-phishing attacks, etc. [28].

In recent years Vendor-created/Open-source threat intelligence sharing platforms, have become a popular choice within industry. These platforms provide organisations with an environment where they can share and consume CTI in either a fully or semi automated way. During their 2021 survey the SANS institute noted that these types of sharing platforms saw a 3% increase in use compared to 2020 [6]. Moreover, it was also reported that more traditional sharing mechanisms (e.g., emails and briefs) saw a 7.8% decrease in use compared to 2020.

We argue that while this trend towards either fully or semi automated threat intelligence sharing is positive, a number of key challenges (e.g., Produce Consumer Imbalance, Data Validity) are currently prevalent in this space, as highlighted by existing literature [43]. Furthermore, we also seek to provide a unique insight into how privacy, trust and accountability define a seemingly paradoxical relationship. As well as discussing several general CTI sharing challenges, we also seek to demonstrate that using a decentralised platform for CTI sharing between organisations in a trustless manner has tremendous promise.

Towards this, blockchain is a promising technology. Blockchain is a tamper-proof, decentralised, and immutable storage of digital information that is impossible to change [24]. Therefore, blockchain can provide a strong and effective solution for securing CTI in networked ledgers, a series of blocks that are cryptographically linked, and facilitates secure dissemination between organisations. However, blockchain-based CTI sharing solutions are lacking in the present literature. A few proposals, e.g., [4, 11, 12], integrate blockchain for CTI sharing, but a comprehensive solution which addresses all of the discussed challenges is currently lacking.

In this chapter, we evaluate a number of present CTI sharing challenges and discuss how blockchain can bring opportunities to address these challenges. Thus, the major contribution of this chapter is to provide a list of challenges associated with CTI sharing and deliver a list of opportunities present within the blockchain space for future research.

The remainder of the Chapter is organised as follows. In Sect. 2, we present a brief overview of blockchain and CTI. In Sect. 3, we present a simplified blockchain-based CTI sharing model from the current literature to demonstrate how blockchain can facilitate sharing. In Sect. 4, we discuss the a number of challenges associated with CTI sharing. In Sect. 5, we present a number of opportunities that highlight the applicability of blockchain-based models in the CTI sharing space based on current ideas presented in the literature. In Sect. 6, we present a brief discussion the related work within the literature. To concluded, in Sect. 7 we summarise the work presented in this Chapter and discuss future research directions.

## 2 Overview of Blockchain and CTI

Before discussing blockchain-based CTI sharing in detail, we present a brief overview of blockchain and CTI in this section.

### *2.1 Blockchain*

Blockchain is a distributed digital ledger for storing electronic records [24]. In other words, blockchain can be seen as a network of computers that store transactions (and therefore the data) across multiple computers. These computers are considered a node in the blockchain. The data entered in a particular interval in the chain is known as a *block*. Each block is identified using a unique identifier is called a *hash*. Each block contains the hash of the previous block. A hash is the output of a unique cryptographic function that takes as input a arbitrary amount amount of data and generates a fixed-size output, the hash. Significantly, this is a one-way function and it is impossible to reserve the computation [44].

In blockchain, when a transaction is first equested, it is authenticated using cryptographic keys (public and private keys). Then a block containing that transaction is created and sent to the entire network. Once the transaction is agreed between the nodes in the network, it is approved (i.e., authorised) before the block is added to the chain. This is done by a mechanism called *consensus*, where the majority of nodes agree with the transaction. Note that nodes must perform a complex mathematical problem to validate a transaction. This is known as *mining*, and the participanting nodes are referred to as the *miners*. Commonly, the mining task in called *Proof of*

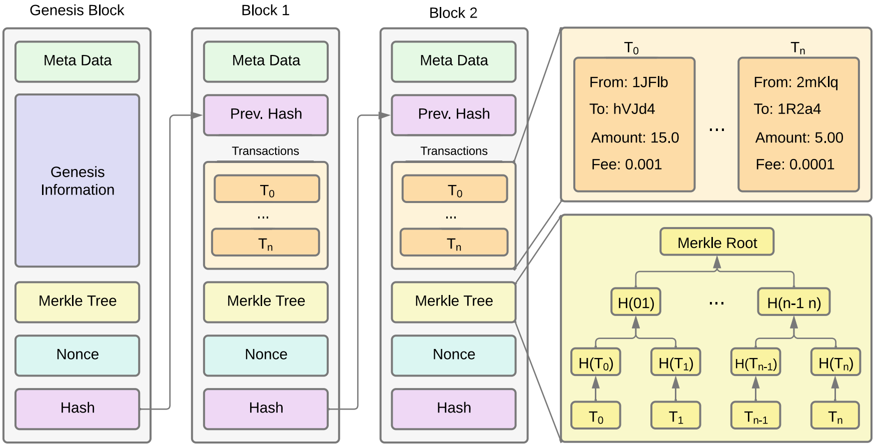


**Fig. 1** An illustration of blockchain immutability created by hashing

*Work (PoW)*. A cryptocurrency reward is given to the miner who first solves the mathematical problem (i.e., the PoW) and validates a block. After this, the block is added to the existing chain, and all the nodes in the network are updated with this information [45].

Therefore, blockchain provides a framework in which nodes can maintain an immutable ledger of data. In Fig. 1, we illustrate how immutability is created in blockchain by linking successive blocks together using cryptographic hashing functions. Currently PoW is the most widely used mechanism for mining. However, PoW requires a substantial computing power and therefore uses considerable amounts of energy, a notable drawbacks of PoW. To solve this issue, another mining mechanism, *Proof-of-Stake (PoS)* is becoming popular. PoS provides faster transactions and uses less energy during mining [19]. Some significant properties of blockchain are outlined as follows [9]:

- *Decentralised*: Does not require a single central authority to validate transactions rather a consensus algorithm is used. As a result, Blockchain does not suffer from a single point of failure.
- *Immutability*: Once a block is added to the chain it is almost impossible to delete or change the data. This provides security to the stored data.
- *Anonymity*: Provides nodes with the ability to participate without disclosing their identity.
- *Trustless*: Nodes also do not have to have pre-established trust with each other to transact, with all transactions documented on the ledger to ensure transparency.
- *Auditability*: At any point in time an existing transaction on the chain can be validated. To ensure a transaction has not been changed or altered the proceeding blocks hashes can be checked.
- *Transparency*: Every transaction that takes place is stored on the blockchain and therefore is visible to the every node in the network.

- *Use of Smart Contracts*: Transaction in the blockchain can be automated with smart contracts. It is a computer code that facilitates and verifies the nodes' agreements and therefore increases the computational efficiencies.

## *2.2 CTI*

Cyber Threat Intelligence refers to a collection of evidence-based knowledge about cyber threats. This knowledge can be compromised of a variety of information including Indicators of Compromise (IoC), attackers' motivations, intentions, characteristics, attack vectors, as well as their Techniques, Tactics, and Procedures (TTPs) [22]. CTI can also consist of actionable advice to detect, prevent, and mitigate the impact of attacks. It can also be obtained from a variety of sources, including anti-virus programs, open-source intelligence (OSINT), Intrusion Detection Systems (IDSs), human intelligence, malware analysis, code repositories, and CTI sharing platforms. CTI can be categorised into following four types: (i) strategic, (ii) operational, (iii) tactical, and (iv) technical. A brief description for each type follows:

- *Strategic* CTI provides a high-level overview of the threat landscape in terms of past, current, and future trends. This type of CTI is often presented in plain language and is focused on improving situational awareness and presenting business risks. The intended audience is senior, lay-person decision makers in an organisation.
- *Operational* CTI refers to information about the nature and motivations of potential upcoming attacks against an organisation, that can be used to formulate targeted prevention strategies and prevent future incidents.
- *Tactical* CTI relates to TTPs and IoCs, that are useful to identify specific attack vectors and vulnerabilities for the purposes of proactively updating signature-based defences against known threats.
- *Technical* CTI consists of technical information often found on threat intelligence feeds about malware and adversarial campaigns, including information about an attacker's assets, attack vectors employed, Command and Control domains used, and types of vulnerabilities exploited.

CTI deals with the collection and analysis of evidence-based knowledge about existing or potential threats that can be used to inform decision making. The aim of CTI is to aggregate a number of unstructured data sources (e.g., network logs and software signatures) and create structured intelligence that details a threat [16].

As noted in Sect. 1, traditional CTI sharing systems lack the ability to share this intelligence effectively. Several of the major challenges that these systems have yet to overcome are—the producer consumer imbalance, data validity, legal and regulatory factors, and sharing intelligent intelligence. Consequently, a number of recently proposed CTI sharing platforms have integrated blockchain into their design to try and provide novel solutions to these challenges.

## 3 Blockchain-Based CTI Sharing

Significant diversity exists in the blockchain-based CTI sharing space. These models utilise specific blockchain characteristics and cryptographic constructs in a variety of ways to facilitate sharing. In Fig. 2, we illustrate a simple sharing framework which exemplifies how blockchain can be applied to CTI sharing [25]. This model is composed of the following components.

- *Consumers*: Users who consume shared CTI information. Make decisions about which intelligence they consume based on the relevance to their physical infrastructure or business case.
- *Producers*: Users who produce CTI based on internal information that can be linked to an existing or new threat. This CTI is then shared with an individual, group or publicly, based on sensitivity of the intelligence using blockchain.
- *Verifiers*: Users who validate shared CTI to ensure it meets sharing standards (e.g., Complies with STIX format), is not duplicated intelligence that has already been shared, and or maliciously contains fake information. The results of this user's analysis either directly impacts the addition of CTI to the blockchain or is added with the given CTI as a report to inform consumer decisions.
- *Authority*: Users who verify the identity of other users before they participate in sharing. This authentication creates trust between users who produce and consume intelligence as they can be sure that only authenticated users are able to do so.

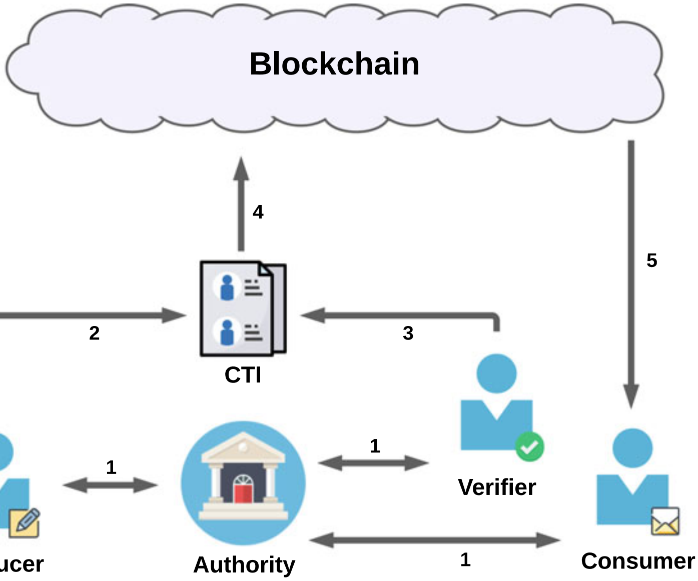


**Fig. 2** A typical blockchain-based CTI sharing framework

- *Blockchain*: It is used to provide a distributed ledger of CTI information (e.g., Hyperledger, Ethereum, EOS, etc).
- *CTI Smart Contract*: Self managed code that is executed by the blockchain to manage the verification of shared CTI. This contract is made up of a Inter Planetary File System reference to the shared CTI and a verification status.

*Note: Users can be any combination of the above roles and subsequently are not restricted to one role.*

As shown in Fig. 2, the process of communication among the various components of the framework follows these steps.

- *Step 1*: All stakeholders prove their identity to a trusted Authority. Proof-of-identity can consist of the exchange of information like government credentials (e.g., drivers licence or passport), ownership of third party certificates or industry accreditation.
- *Step 2*: Producer generates CTI and adds it to the blockchain for verification.
- *Step 3*: Verifier determines the credibility of the CTI based on a set of standards agreed upon by the network.
- *Step 4*: CTI that is determined to be valid in *Step 3*, is added to the blockchain.
- *Step 5*: Consumers access CTI that has been added to the blockchain.

The simplified sharing model presented in Fig. 2 demonstrates how blockchain can be used to facilitate CTI sharing at a basic level. Moreover, when the properties of blockchain discussed in Sect. 2.1 are considered in the context of CTI sharing, the advantages that blockchain-based sharing models have over traditional centralised approaches can be highlighted.

## 4 Challenges

Traditional CTI sharing frameworks (e.g., MISP, OpenCTI and ISACs) have a wide range of challenges that are documented in the literature [42, 43]. In this section, we focus on a subset of these general CTI sharing challenges (cf. Fig. 3).

### *4.1 Producer Consumer Imbalance*

Stakeholder who participate in CTI sharing as either a producer or consumer (cf. Sect. 3) must consider the risks and benefits associated with doing so. In the case where a producer shares intelligence, a number of reputational and or monetary risks are prevalent. For example, sharing intelligence that indicates an organisation has been the victim of a ransomware attack, could cause stock prices to fall or new customers to choose a competitor. Some of the potential risks are listed below [42, 43];

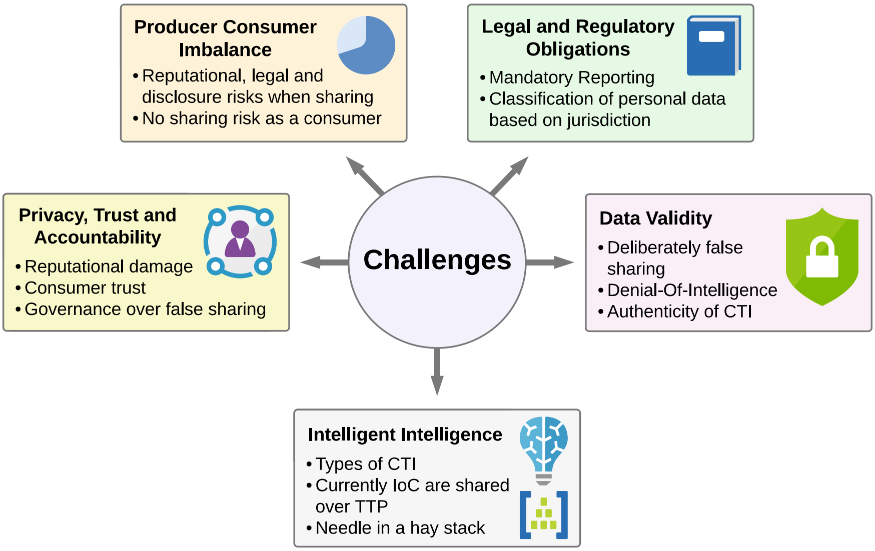

Producer Consumer Imbalance
• Reputational, legal and disclosure risks when sharing
• No sharing risk as a consumer
Legal and Regulatory Obligations
• Mandatory Reporting
• Classification of personal data based on jurisdiction
Privacy, Trust and Accountability
• Reputational damage
• Consumer trust
• Governance over false sharing
Challenges
Data Validity
• Deliberately false sharing
• Denial-Of-Intelligence
• Authenticity of CTI
Intelligent Intelligence
• Types of CTI
• Currently IoC are shared over TTP
• Needle in a hay stack

**Fig. 3** General CTI sharing challenges

- *Consumer Distrust*: Potential consumers might feel that a reported cyber incident means that the organisation is vulnerable. As a result, existing customers may decide to use the services of a competitor that has not reported an incident.
- *Competitor Advantage*: Competitors become aware of potential vulnerabilities that might affect them without being directly affected by it. This allows them to implement mitigation strategies for the same vulnerability at a reduced resource cost.
- *Revealing Trade Secretes*: Information about hardware, software or services an organisation use might be revealed.

Apart from being able to consume CTI themselves, producers do not gain any direct benefits from sharing. As a consequence, without implementation of a reward-based system as part of a sharing platform, the process of sharing CTI can be considered a common good service. On the other hand, consumers assume almost no risk when consuming CTI. Even in the case where the consumption of specific CTI is attributable to an organisation, this action alone is not likely to result in the same reputational or monetary consequences associated with sharing. Given that organisations that consume CTI can implement mitigation strategies against vulnerabilities before they can be acted on, we propose that the following benefits that could be gained;

- *Increased Service Quality*: Increase service up time provides existing customers with a better service quality. This could result in a higher customer retention rate. As a result of providing existing customers with a more consistent service, an organisation might gain a reputation for providing services with low downtime.

– *Reduced Negative Publicity*: In the case where an organisation successfully implements a mitigation strategy to fix a shared vulnerability, the potential for negative publicity due to a successful attack is removed.
– *New Customers*: In the case where an organisation has suffered from a number of cyber incidents (e.g., DDoS Attacks, Privacy Leak), it could be predicted that dissatisfied customers could seek an alternative service. Moreover, if a competing organisation providing and analogous service that has not suffered from these same incidents due to the consumption of CTI, it could be predicted that this organisation could gain additional customers.

From the above discussion it is clear that the risks and benefits associated with the producer and consumer role are not equal. This inequity, consequently creates an imbalance. If this imbalance is not addressed as part of a sharing platforms design, organisations can be observed to exhibit *free-riding* behavior [43]. In this case, *free-riding* behavior can be defined as a deliberate lack of participation by organisations who could share valuable CTI, however choose not to. If a large portion of organisations deliberately behave in this way, the productivity of a sharing platform is affected in two major ways [36];

1. Not sharing removes the ability of other organisations to mitigate against the same incident. When CTI is shared, it is possible for other organisations to put in place mitigation strategies (e.g., Firewall rules) to ensure they are not susceptible to attacks which have a similar profile or share common characteristics. In the case of *free-riding*, this is not possible.
2. *Non-free-riding* organisations might stop or reduce the amount of intelligence they share due to a lack of consumable CTI from others. As noted above, producers assume a number of risks when they participate in sharing. However, if part of a productive platform where a large volume of valuable intelligence is shared, this risk compared to the benefit gained by consuming other intelligence makes sharing more attractive. Consequently, a large portion of *free-riding* organisations has the potential to impact the sharing behaviours of others.

## *4.2 Legal and Regulatory Obligations*

Organisations who participate in sharing have to follow the legal and regulatory obligations associated with the jurisdiction they are from. Survey [43], highlights a number of legal and regulatory obligations that organisation in certain countries must meet.

For example, in Germany Internet Protocol (IP) addresses are considered personal information and therefore any disclosure of CTI containing them must comply with German privacy laws [26]. However, in the UK IP addresses are not considered personal information and therefore can be freely shared. In terms of CTI sharing, IP addresses are likely to be shared as an IoC and therefore organisations based in these different jurisdictions have to ensure they comply with the applicable laws.

Moreover, countries like Belgium and Slovenia have mandatory sharing legislation [43]. This legislation requires organisations from these two countries to report any cyber incidents to a specific authority when they occur. If these organisations were also to participate in CTI sharing on top of this, in some cases the resources consumed to facilitate both of these independent sharing requirements could exceed those which are available.

The above examples highlight that while theoretically CTI sharing is ubiquitous across the world, legal and regulatory obligations can pose a significant barrier. Given that legal and regulatory obligations are significantly diverse across the world, sharing platforms must ensure CTI can be shared in a pliable way.

## *4.3 Data Validity*

Threat hunting is defined by [7, 14] as the proactive approach of seeking anomalous or malicious activity within an organisations cyber terrain. The process of performing this task, which if successful can result in the production of CTI, can be highly variable in nature. At the most basic level, threat hunting can simply consist of manual analysis of network or Windows logs. In contrast, [3] proposes a sophisticated threat hunting model which utilises machine learning to automatically generate threat intelligence based on data from a variety of sources.

While these examples vary in their sophistication, they both share the common feature that the process of generating CTI is solely completed by the sharing organisation. As a result, it is possible for malicious organisations or individuals to intentionally generate and share false intelligence. We note that sharing false CTI has the potential to be utilised in several ways to either gain additional attack surfaces or to bury real CTI amongst fake intelligence going forward. Two examples of this are discussed below.

**Automatic Attack Feed Exploitation**: Recent trends in CTI sharing have seen many notable developments towards automation, both in its generation (as discussed above) as well as in its consumption. For example, technologies such as Structured Threat Information Expression (STIX) and Trusted Automated eXchange of Intelligence Information (TAXII), has allowed many organisations to easily share and consume CTI in an automated way [43]. As CTI consumption becomes more automated, it could be feasible for threat actors to utilize this to create new attack surfaces. For example, intelligence structured using STIX can contain SNORT rules that consumers can automatically feed into their intrusion detection systems (IDS) [42]. Given certain conditions, we theorise that it could be plausible for an attacker to construct seemingly legitimate intelligence that causes a consuming organisations IDS to flag legitimate activity as malicious. This technique could be used in conjunction with an actual attack, to disguise malicious activity amongst legitimate traffic that is falsely flagged as suspicious.

**Denial-of-Intelligence**: As sharing platforms become more and more effective at allowing organisations to mitigate against threats, they themselves could become targets. Denial-of-Service (DoS) attacks have been around since the origin of the internet yet still remain highly effective in the present day. The main goal of a DoS attack is to simply make a particular computing resource unavailable [8]. The most common way that these types of attacks are committed, is by overwhelming a service with a large volume of bogus requests. We observe that a ‘Denial-of-X’ style attack could be constructed to target CTI sharing platforms specifically.

In this case, threat actors could develop Denial-of-Intelligence (DoI) attacks. This type of attack would seek to overwhelm a platform with a large amount of bogus intelligence. By flooding a sharing platform with a large amount volume of fake intelligence, threat actors could exploit a common vulnerability across multiple targets. The result of this would mean that while valid intelligence detailing the attack could be shared by the initial victim, it is buried amongst an overwhelmingly large volume of the false information.

Both the above examples highlight that the ability to accurately determine the validity of shared CTI is a critical challenge that platforms must find novel ways overcome. Moreover, these examples also indicate that as the process of sharing becomes more automatic and widely used, data validity becomes more critical.

### *4.4 Intelligent Intelligence*

In Sect. 2.2, we discussed what CTI is and highlighted that it can be categorised into four main types: (i) strategic, (ii) operational, (iii) tactical, and (iv) technical. Each of these types of intelligence convey a narrative about a threat, however do so in diverse range of ways, specific to their intended recipients. For example, Technical CTI is made up of data that describes the physical attributes of an observed attack (e.g., IP address, MAC address, Malware Hash, etc), intended to be consumed by technical resources [42].

It is important to understand that these types of intelligence are highly variable in their sophistication. In this case, sophistication refers not just to the quality of the CTI itself, but how consumers are able to use it. Proposal [22] makes an important distinction between data, information, and intelligence, that highlights this variability. They are as follows:

- *Data*: Simple facts that can be made available in large volumes such as IP addresses, logs, hashes.
- *Information*: A collection of raw data together that shows suspicious activity.
- *Intelligence*: The process of analyzing and drawing meaningful conclusions that can be used by security professionals to define an intelligence-lead approach to decision making.

If the above criteria are applied to the categories of CTI discussed in Sect. 2.2, tactical, operational and strategic CTI could be classed as intelligent intelligence. On the other hand, technical intelligence (e.g. IoC) can only be classified as data/information intelligence, and subsequently cannot directly inform decision making. As a result, intelligence types can be grossly defined into high-level intelligence (e.g., TTP) and low-level intelligence (e.g. IoC).

Currently, the majority of exchanged CTI can be classified as low-level intelligence [1, 16, 38]. Survey [42], notes that over 250 million IoC are shared cumulatively across CTI sharing platforms every day, with this figure likely increasing in recent years. From the outset, this trend of sharing large volumes of technical intelligence may appear positive. However, when framed from the perspective of a consuming organisation, the quantity of available intelligence becomes an interpretability challenge analogous to the *needle in a haystack* problem.

## 4.5 Privacy, Trust, and Accountability

Privacy, Trust and Accountability, are three factors that any CTI sharing platform must balance to facilitate an environment conducive to share and consume CTI [2, 27, 43]. The relationship between each of these factors and CTI sharing are discussed below;

**Privacy** can be defined as the ability or inability for a consuming organisations to associate some shared intelligence with the sharing organisations real identity. The literature consistently highlights reputational damage as a significant barrier that stops organisations from participating in CTI sharing [1, 42, 43]. Given that reputational damage can result from sharing intelligence in an identifiable way, a degree of anonymity is required when sharing.

**Trust** can be defined as a consumers ability to trust the intelligence which they receive [38]. Subsequently, a trust relationship between CTI producers and consumers is present in any sharing platform. In contrast to privacy, the parameters used to define the trust relationship between producers and consumers often require some link to the producer's real identity. By linking at some level a producers real identity to the CTI they share, consumers have greater assurance that shared intelligence comes from an authoritative source [39].

**Accountability** can be defined as the ability for a sharing platform to provide governance shared CTI. In this case, Governance refers to a sharing platforms ability to hold users who participate in false sharing responsible. Subsequently, the ability to hold users accountable for their actions insures that the integrity of shared intelligence can be maintained. Like trust, accountability is also dependent on being able to reveal a producers real identity given that they have made a malicious contribution [20, 40].

From the above discussion, it can be hypothesised that privacy, trust, and accountability form a paradoxical relationship. Producers of intelligence want to be completely anonymous when sharing. However, it is the preference of CTI consumers to have proof that the intelligence they consume originates from a reputable source [23].

Moreover, the group of users who make up a sharing platform should have governance over the information shared, and consequently be able to hold users who share false information accountable. As a result, the way in which CTI sharing platforms manage privacy, trust, and accountability is an important challenge.

# 5 Opportunities

In this section, we discuss a list of opportunities (cf. Fig. 4) for blockchain-based CTI sharing. These opportunities aim to highlight how the characteristics of blockchain can be leveraged to provide novel solutions to the challenges discussed in Sect. 4.

## *5.1 Incentivised Sharing*

To help alleviate the producer consumer imbalance discussed in Sect. 4.1, several incentive schemes can be implemented. In this section we will discuss two examples that illustrate how incentivised sharing can be achieved using blockchain.

**Concessions**: Some blockchain-based sharing platform, such as [25], use subscription fees to create permissioned sharing groups. Consequently, users are required to pay an authority a subscription fee to participate, consume and or share CTI, for a

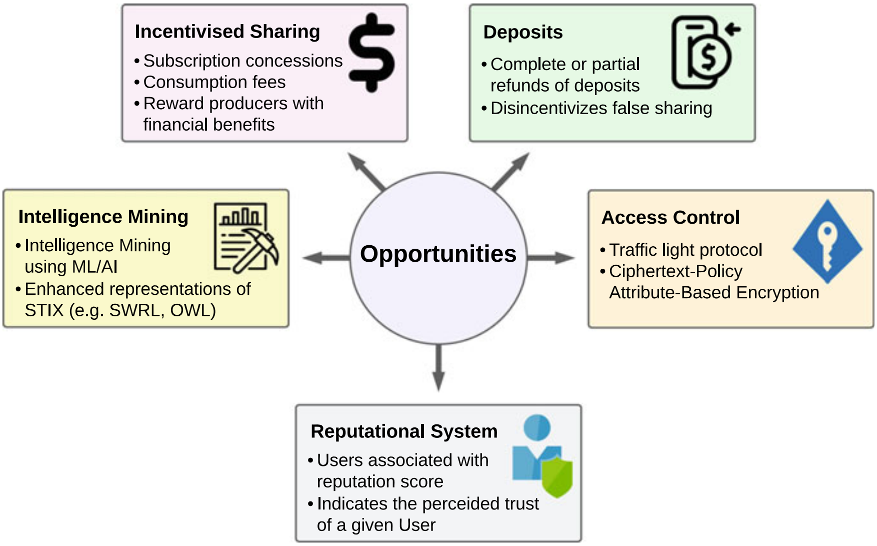

Incentivised Sharing
• Subscription concessions
• Consumption fees
• Reward producers with financial benefits
Deposits
• Complete or partial refunds of deposits
• Disincentivizes false sharing
Intelligence Mining
• Intelligence Mining using ML/AI
• Enhanced representations of STIX (e.g. SWRL, OWL)
Opportunities
Access Control
• Traffic light protocol
• Ciphertext-Policy Attribute-Based Encryption
Reputational System
• Users associated with reputation score
• Indicates the perceided trust of a given User

**Fig. 4** Blockchain-based CTI sharing opportunities

given time period. To incentivise users to share CTI and not just consume it, concessions can be given to users who contribute intelligence. As a users contributions are stored using blockchain (e.g. In a Smart Contract or directly on-chain), an auditable and immutable record of these transactions is maintained. This record can thereafter be used by an authority to determine a users subscription fee once their previous subscription has expired. In the case were a users record shows that they have shared valuable intelligence, the price of their next subscription can be lowered to incentives them to continue making valuable contributions going forward.

An example of a sharing model that implements concession based incentives is [25]. In this model, the authors use subscription discounts to reward CTI producers for their contributions. As part of their implementation, proposal [25] provides a CTI producer with a discount each time they share intelligence that is considered high-quality by a set of verifiers. This design therefore allows users who continually share high-quality CTI to significantly reduce their subscription fees. To achieve this, CTI sharing is completed using the following steps;

1. CTI producer adds CTI to the blockchain.
2. Three random verifiers are selected from a trusted group.
3. Verifiers rate the CTI's quality using pre-determined metrics.
4. If the majority of verifiers rate the given CTI as high-quality then both the producer and verifiers are given a discount on their next subscription.
5. If the majority of verifiers rate the given CTI as low-quality then only the verifiers are given a discount on their next subscription.

**Fees**: Another example of how blockchain can be used to combat free riding behavior within CTI sharing is consumption fees. Unlike subscription concessions, consumption fees require consumers to pay producers to access CTI that they have shared [21, 35]. In essence, consumption fees aim to create a market place where CTI can be exchanged between organisations for currency. Due to the trustless properties of blockchain, CTI can be exchanged between two organisations without the need to pre-establish trust or use a third party. Instead, self manged Smart Contract can be used to facilitate the exchange of CTI and cryptocurrency between two organisations. By creating a blockchain-based CTI marketplace, producers who actively share valuable CTI have the ability to profit significantly from doing so.

Two main approaches can be used to implement consumption fees within blockchain-based architectures.

1. **Standard fee**: A predefined fee is payed to producers when other organisations consume their CTI [35].
2. **User defined fee**: Producers specify a consumption fee which is payed each time a user access it [21]. Can be implemented as a producer defined parameter in a Smart Contract.

Decentralised incEntives for threAt inteLligEnce Reporting and exchange (DEALER), is an example of a blockchain-based CTI sharing platform that implements a user defined consumption fee to incentivise CTI sharing [21]. The below steps summarise how DEALER implementes user defined consumption fees.

1. CTI producer adds CTI to the blockchain. During this process, the producer can define a sale price. If a producer does specify a sale price, they also have to pay a verification fee.
2. In the case where a sale price is specified, three trusted verifiers review the associated intelligence using pre-determined metrics.
3. The results of each verifiers analysis are added to the blockchain to indicate to buyers the quality of the given intelligence. Moreover, each verifier is given a proportion of the verification fee.
4. When a buyer purchases some intelligence they are required to pay the associated consumption fee, if specified by the producer.

As discussed in Sect. 4.1, an imbalance between the producer and consumers roles exists within CTI sharing. Consequently, it is critical that sharing platforms seek to address this imbalance by providing producers with more direct benefits. In this section we highlighted a number of way in which blockchain-based sharing platforms can implement different incentive schemes to combat the effects of the producer consumer imbalance.

## 5.2 *Deposits*

In Sect. 4.3 the issue of false sharing was discussed. To disincentivise CTI producers from participating in this behaviour negative financial punishments can be used. In the case of blockchain-based platforms, existing technologies that support the exchange of cryptocurrency can be utilised for this purposes (e.g. Ethereum). Moreover, many of these platforms also allow self managed Smart Contracts to exchange cryptocurrency autonomously, thus removing the need for a trusted third party [41]. As a result, Smart Contracts can be utilised to implement conditionally refundable deposits in a trustless, auditable and verifiable manner.

Conditionally refundable deposits can be utilised by blockchain-based CTI sharing platforms to introduce negative financial punishments for CTI producers that participate in false sharing. In this case, when a producer shares some intelligence they could be required to pay a deposit, some amount of cryptocurrnecy, to a Smart Contract. Once payed, a consensus algorithm defined within the Smart Contract can be used to verify the integrity of the shared intelligence [11]. Given that this verification process occurs on-chain, the results are immutable and transparent to both the original producer as well as future consumers. Furthermore, the autonomous and deterministic nature of Smart Contracts allows them to hold cryptocurrency in escrow without the need for pre-established trust.

In the case where shared intelligence is found to be credible, the initial deposit can be payed back either in full or partially to the original producer automatically by the Smart Contract. On the other hand, when false sharing is found to have occurred this deposit can either be held by the Smart Contract, burned or distributed to users involved in the verification process [41]. By punishing users who participate in false sharing, persistent efforts to do so on a large scale are deterred due to the associated financial cost.

**BLOCIS**: In [11], the authors use conditionally refundable deposit to disincentivise stakeholders from deliberately sharing false/incorrect CTI. When a registered BLOCIS user shares CTI, they use a pre-defined Data Report Contract (DRC). This contract takes as input the given CTI as well as a deposit. Once added to a specific feed, an evaluation function ($\pi$) is used to assess the validity of the reported intelligence. The novelty that BLOCIS proposes is that $\pi$ takes as input both the reputational score of the producer as well as their deposit. If the output of $\pi$ indicates the given CTI is false, the deposit is not refunded back to the producer. When simulated in a test environment, the BLOCIS model was found to successfully disincentivise users who made malicious contributions. Figure 5 in [11] demonstrates both the financial and reputational damage that users who participated in false sharing suffered over an extended period of time.

**Considerations**: While deposit-based disincentive schemes are focused on punishing malicious producers, considerations must also be made to ensure honest producers are not deterred from sharing. Although the self managed nature of Smart Contracts can provide producers with a trustless way to exchange cryptocurrency, factors such as the amount of currency and consensus used to determine if a contribution is false must be considered. For example, if producers are required to pay a constant amount of cryptocurrency, the extremely volatile nature of currency markets could cause producers not to share at particular times [37]. Moreover, if consensus methods are dependent on validation of intelligence from a set of validators, then they themselves could become by subject to malicious attacks. Given cryptocurrency is at stake, we argue that malicious attacks could seek to compromise a subset of validators to deny the authentication of any intelligence. Lastly, if validators are directly incentivised through partial payment of deposits from intelligence deemed malicious, then validators might be more likely to classify honest contributions as malicious. All of these factors need to be considered carefully when designing a deposit-based disincentive scheme as they have the potential to affect honest producers as well.

Aside from considerations related to how deposit-based Smart Contracts are designed, the method used to validate intelligence is another important factor. Fundamental to the success of conditionally refundable deposits is the ability of a verifier or group of verifiers to determine the credibility of CTI. However, currently a method that deterministically classifies CTI as false is considered an open challenge [11]. Consequently, platforms that implement disincentive schemes are likely to encounter cases where CTI is wrongly considered malicious and an honest producer is punished.

### 5.3 Reputational System

Another way blockchain-based solutions can mitigate against false sharing is use of reputational systems. Unlike deposits, reputational systems do not punish malicious users monetarily. Instead, they associate each user with a reputation score (e.g. 1–100) that represents their perceived trustworthiness.

In the context of CTI sharing, a users reputational score can be used to directly affect their ability to consume and or share intelligence within a group [11, 46]. For example, if a CTI producer shares some CTI, their associated reputational score could be used to indicate to validators and or consumers the level to which they should trust it [11]. As a result, intelligence shared by a users with a relatively low reptuational score might be subject to more thorough inspection by validators. In the opposite case, users who have a relatively high reputational score, may be subject to less thorough inspection by validators. Furthermore, these highly trusted users might be able to consume more sensitive intelligence that might otherwise have been unavailable to them.

A successful reputational system has the potential to stop a user or group of users from continually false sharing [11]. Given that a users reputational score is tightly coupled with their perceived trust, efforts to continually false share can be predicted to become harder over time.

**Proof-of-Reputation (PoR)** is a blockchain-based consensus algorithm that was proposed by [46] specifically for CTI sharing. In their model, each node in the network has an associated reputational score between 1 and 100. Fundamentally, this score seeks to capture how trustworthy a user is based on the creditably of their previous contributions. Importantly, all of the actions taken by a node (e.g. Voting, Sharing CTI) influence its reputational score over time.

When an organisation shares CTI, other nodes on the network calculate a reputation value which is used to judge if it should be added to the blockchain. The results of this reputation-based consensus are used alongside more traditional validation methods to try and mitigate against false sharing. Moreover, a contributing nodes reputational score is adjusted over time based on the results of this process. Critical to the integrity of this process is a predefined trust threshold. This trust threshold defines the point at which a node is considered trustworthy. As a result, if a nodes score drops below this threshold, then it is considered untrustworthy and cannot participate further.

The above PoR consensus algorithm exemplifies how the inherent properties of blockchain discussed in Sect. 2.1, can be utilised to facilitate reputational systems without the need for a trusted third party. In particular the immutable, transparent and auditable properties of blockchain allow each node to calculate the reputational scores of others, thus removing the need for a centralised authority. Similar to [46], the BLOCIS architecture proposed by [11] manages reputational scores with self managed Smart Contracts. Like deposits, these Smart Contracts contain a predefined consensus algorithm that can be used to manage the reputational scores of each user over time in a trustless way.

As mentioned in the in Sect. 5.2, the ability to deterministically validate CTI is still an open challenge. Given that reputational systems require a verifier or group of verifiers to determine the credibility of CTI, their success is dependent on the accuracy of the validation method used.

### *5.4 Access Control*

Blockchain-based sharing platforms can use several methods to provide producers with control over who has access to the intelligence they share. Access control in this case, refers not just to the ability of CTI producers to control who has access to their intelligence, but also in what way [30–32]. For example, a particular producer might want to share CTI with a small trusted group. However, they only want to disclose the specific attribute values (e.g., IP addresses) associated with it to one of the organisations.

While access control can be implemented by centralised architecture, blockchain is able to facilitate the fine grained access control required for CTI sharing in a trustless way. The following list outlines how a number of the key properties of blockchain can be leveraged to provide access control in a trustless way.

- *Decentralised*: As a single authority does not control access based on a producers request, greater integrity is achieved. This means that producers have greater confidence that the control policy they define will be followed given its execution is not dependent on a centralised system.
- *Immutability*: CTI producers can be confident that the access control parameters they define cannot be altered by another user for their benefit.
- *Smart Contracts*: Provides a framework to allow stakeholders to define the access control for the intelligence they share. Moreover, the self managed nature of Smart Contracts ensure that these access control policies are executed autonomously.

**Traffic Light Protocol (TLP)** is an example of an access control method that can be implemented as part of a blockchain-based sharing platform [25]. TLP defines a robust access control structure that gives producers the ability to specify who CTI is shared with. This is achieved by allowing producers to specify a sharing level from a predefined list. Each of these predefined sharing levels is simply a control policy that specifies which users can access the CTI. Table 1 is an example of how a TLP policy could be structured.

**Ciphertext-Policy Attribute-Based Encryption (CP-ABE)** is another method that can be used to give producers with fine grained access control [33]. In the case of CP-ABE, when a producer shares CTI, they encrypt it using attribute-based encryption methods [5]. The ciphertext that results from this process is then added to the blockchain. When a user access this ciphertext they are able to decrypt parts of it based on their own attributes. As a result, users can define highly specific fine grained access control policies using CP-ABE.

**Table 1** Example of a TLP implementation by [12, 25]

| Channel | Description |
|---|---|
| Red | Private channel between two stakeholders only |
| Orange | Disclosure to only a certain group of stakeholders defined by the CTI producer |
| Green | Disclosure to an entire group of stakeholders. In the case of private blockchain this is restricted to anyone who has access to it |
| White | Public disclosure which is accessible to anyone |

For example, a CP-ABE policy might require that an organisation is a *ICS-ISAC* member to view a subset of the CTI. Furthermore, it might also specify that only a specific subset of these organisations can access the specific details of the hardware affected by a ransomware attack. This example demonstrates how CP-ABE can be used to construct fine grained access control policies specific to a producers needs.

Both TLP and CP-ABE are examples of access control methods that can be implemented using blockchain. Importantly, these methods provide CTI producers with better control over who consumes the intelligence they share in a trustless way. In Sect. 4.5, the issue of privacy was discussed. During this discussion, it was highlighted that fear of reputational damage was a significant barrier that stopped some organisations from sharing. While greater access control does not provide a complete solution to this problem, we argue it has the potential to cause more organisations to share within closed groups given their privacy-preserving nature. Moreover, if key regulatory bodies are incorporated into sharing platforms, these frameworks can further help organisations meet their legal and regulatory obligations without having to use secondary sharing mechanisms [25].

## 5.5 *Intelligence Mining*

In Sect. 4.4, it is noted that not all types of CTI are equivalent in their ability to describe threats and subsequently be used to implement mitigation strategies against them. Given that the process of generating CTI is dependent on the capabilities of the sharing organisation, it cannot be expected that all organisations are capable of generating high-level intelligence. As a result, strategies to create high-level intelligence from aggregated sources of low-level intelligence have the potential to shift sharing towards more intelligent intelligence. Furthermore, this process also allows organisations which do not have the resources to generate high-level intelligence themselves to still contribute.

Intelligence mining can be defined as the process of deriving high-level intelligence from low-level intelligence already stored on the blockchain [42]. The immutable and auditable properties of blockchain are able to facilitate mining in a trustless way. Given that low-level intelligence used as part of the mining process is immutable and accessible by each organisation on the network, high-level intelligence that is derived from it can be validated by other organisations. As a result, the ability to mine high-level intelligence in a trustless way has the potential to allow blockchain-based CTI sharing platforms to provide participating organisations with more advanced threat mitigation.

Proposal [35] provides an example of how STIX, Semantic Rule Language (SWRL) and Web Ontology Language (OWL) can be combined to create more meaningful and interpretable representations of CTI. The use of these tools together has great potential in the area of intelligence mining, as CTI represented in this way allows semantic reasoners to infer new knowledge [35]. Furthermore, extending traditional representations of CTI could also pave the way for Machine Learning (ML)/Artificial Intelligence (AI) approaches to intelligence mining. In [13], it was demonstrated that ML algorithms were able to generate CTI from a single organisations network logs stored using blockchain. Therefore, it could be possible to extend this approach further to generate more intelligent intelligence, from large amounts of aggregated CTI expressed using STIX, SWRL and OWL.

## 6 Related Work and Discussion

In this section, we present some related works on CTI sharing and the integration of blockchain platforms for CTI sharing and provide a discussion on the findings of this chapter. Several studies discuss the importance of CTI sharing in information security and general computing systems [10, 16, 42, 43]. However, most of them discuss CTI sharing from the lens of traditional centralised computing approaches. Subsequently, few publications considering how blockchain-based approaches can overcome existing challenges are present in the literature. In this section, we aim to discuss the contributions of several publications that outline challenges associated with CTI sharing.

Proposal [43], provides a comprehensive insight into what CTI sharing is and how it is commonly performed. Furthermore, it also discusses a number of important CTI sharing concepts including—what CTI is, how it can be shared, and most notably what benefits and risks are associated with sharing. Of particular note, the authors highlight the importance of privacy and anonymity in CTI sharing. However unlike [43], in this chapter, we extended these ideas to consider the relationship between privacy, trust, and accountability.

In [42], a survey on technical threat intelligence was conducted. Like [42, 43] provides a good insight into the key concepts which define CTI sharing. This paper specifically seeks to provide a clear definition of what threat intelligence is and what some of the associated challenges in this space are. An important challenge highlighted by [42], is intelligent intelligence. Moreover, their suggestion that big data analysis could be applied to threat intelligence was extended by our work to focus on how these concepts can be applied to blockchain specifically.

Proposal [38] provides a comprehensive study into the current challenges associated with CTI sharing platforms (e.g. MISP). As part of their research, they investigate twenty two sharing platforms and derived a list of eight key findings. A number of which are discussed in Sect. 4. While their research was mostly focused on centralised architectures, their insights into existing challenges allowed us to highlight how blockchain-based architectures can provide novel solutions to them.

In [1], the authors perform a comprehensive literature review into the current use CTI. As part of their findings, they outline four main challenges of which three were discussed in this chapter. However, unlike our approach, this research does not explore how blockchain-based solutions can provide novel solutions to these challenges.

Recently, a diverse range of blockchain-based CTI sharing models have been published. In this chapter, we discussed a number of novel features present within a subset of these models which we feel represent the current state-of-the-art.

We argue that [35] currently presents the most comprehensive blockchain-based CTI sharing platform, as it addresses a number of the challenges presented in this chapter. As part of their model, the authors integrate a number of features which address the producer consumer imbalance, intelligence intelligence, and legal and regulatory factors. However, it must be noted that while this model does provide trust and accountability, it is achieved at the cost of privacy-preserving anonymity.

DEALER is a blockchain-based CTI sharing platform presented by [21], which like [35], presents novel solutions to a number of the challenges discussed in this chapter. The DEALER proposal provides solutions to the producer consumer imbalance and legal and regulatory factors. Moreover, this proposal also integrates a quality assurance method which provides a heuristic approach to solving the challenge of data validity. It must be noted, however, that while a heuristic approach to the issue of data validity has the potential to be effective, it does not completely mitigate against false sharing.

Few models present in the current literature provide a robust framework that balances privacy, trust, and accountability, as defined in Sect. 4.5. We argue that [2] presents the most comprehensive approach to balancing these factors. The authors of this platform propose a framework which allows CTI producers to share intelligence semi-anonymously while still facilitating trust and accountability. However, the major limitation of this framework is that a single trusted authority has the ability to reveal the identity of any CTI producer, subsequently creating a single point of failure.

We find there are various challenges in CTI sharing, and blockchain is a promising solution to gain opportunities in most cases. However, there is still a list of open research questions that need to be resolved. We list a few of them as follows:

- How the properties of blockchain and other cryptographic constructs be used to create a blockchain-based CTI sharing model that provides a balance between privacy, trust, and accountability?

- How can shared CTI be deterministically validated to ensure false sharing is not possible?
- How can ML/AI be utilised along side current approaches (e.g. STIX, SWRL, OWL) to facilitate the sharing of more intelligent intelligence?

## 7 Conclusion

The drastic evolution of the threat landscape, brought about by the emergence of Internet of Things (IoT) technology, has caused organisations to find new ways to better manage their cyber risks. This appetite for tools that better mitigate against potential threats has driven the development for a number of Cyber Threat Intelligence (CTI) sharing platforms (e.g., MISP). In this chapter, we defined a number of general CTI sharing challenges including the producer consumer imbalance, legal and regulator factors, intelligent intelligence, data validity, and privacy, trust and accountability. These general CTI sharing challenges were then used to deliver a list of opportunities present within the blockchain-based space. These opportunities included deposits, access control, reputational systems, intelligence mining and incentivised sharing. Finally, we explored several existing proposals and determine a list of unique future research questions for efficient and secure CTI sharing using blockchain.

**Acknowledgements** The authors acknowledge the support of the Commonwealth of Australia and Cyber Security Research Centre Limited.

## References

1. M.S. Abu, S.R. Selamat, A. Ariffin, R. Yusof, Cyber threat intelligence-issue and challenges. Indones. J. Electr. Eng. Comput. Sci. **10**(1), 371–379 (2018)
2. Y. Allouche, N. Tapas, F. Longo, A. Shabtai, Y. Wolfsthal, Trade: trusted anonymous data exchange: Threat sharing using blockchain technology (2021). arXiv preprint arXiv:2103.13158
3. M. Arafune, S. Rajalakshmi, L. Jaldon, Z. Jadidi, S. Pal, E. Foo, N. Venkatachalam, Design and development of automated threat hunting in industrial control systems. *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom)*. pp. 618–623 (2022)
4. S. Badsha, I. Vakilinia, S. Sengupta, Blocynfo-share: Blockchain based cybersecurity information sharing with fine grained access control, in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE, 2020), pp. 0317–0323
5. J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in *2007 IEEE Symposium on Security and Privacy (SP'07)* (IEEE, 2007), pp. 321–334
6. R. Brown, R.M. Lee, 2021 sans cyber threat intelligence (CTI) survey, in Tech. Rep. SANS Institute (2021)
7. J.R. Bynum, Cyber threat hunting, Ph.D. thesis, Utica College (2019)
8. E. Chou, *Distributed Denial of Service (DDoS)*, 1st edn. (O'Reilly Media, Inc., 2018)

9. H.N. Dai, Z. Zheng, Y. Zhang, Blockchain for internet of things: a survey. IEEE Internet Things J. **6**(5), 8076–8094 (2019)
10. N.X. Gong, Barriers and impacts to adopting interoperability standards for cyber threat intelligence sharing: a mixed methods study, Ph.D. thesis (Robert Morris University, 2017)
11. S. Gong, C. Lee, Blocis: blockchain-based cyber threat intelligence sharing framework for sybil-resistance. Electronics **9**(3), 521 (2020)
12. D. Homan, I. Shiel, C. Thorpe, A new network model for cyber threat intelligence sharing using blockchain technology, in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (IEEE, 2019), pp. 1–6
13. Z. Jadidi, A. Dorri, R. Jurdak, C. Fidge, Securing manufacturing using blockchain, in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (IEEE, 2020), pp. 1920–1925
14. Z. Jadidi, Y. Lu, A threat hunting framework for industrial control systems. IEEE Access **9**, 164118–164130 (2021)
15. C. Johnson, L. Badger, D. Waltermire, J. Snyder, C. Skorupka, et al., Guide to cyber threat information sharing. NIST Spec. Publ. **800**(150) (2016)
16. H. Kure, S. Islam, Cyber threat intelligence for improving cybersecurity and risk management in critical infrastructure. JUCS-J. Univ. Comput. Sci. **25**, 1478 (2019)
17. H.I. Kure, S. Islam, H. Mouratidis, An integrated cyber security risk management framework and risk predication for the critical infrastructure protection. Neural Comput. Appl. 1–31 (2022)
18. A. Lamssaggad, N. Benamar, A.S. Hafid, M. Msahli, A survey on the current security landscape of intelligent transportation systems. IEEE Access **9**, 9180–9208 (2021)
19. C. Lepore, M. Ceria, A. Visconti, U.P. Rao, K.A. Shah, L. Zanolini, A survey on blockchain consensus with a performance comparison of PoW, PoS and pure PoS. Mathematics **8**(10), 1782 (2020)
20. W. Lueks, M.H. Everts, J.H. Hoepman, Revocable privacy: principles, use cases, and technologies, in *Annual Privacy Forum* (Springer, 2015), pp. 124–143
21. F. Menges, B. Putz, G. Pernul, Dealer: decentralized incentives for threat intelligence reporting and exchange. Int. J. Inf. Secur. **20**(5), 741–761 (2021)
22. J. Moubarak, C. Bassil, J. Antoun, On the dissemination of cyber threat intelligence through hyperledger, in *2021 17th International Conference on the Design of Reliable Communication Networks (DRCN)* (IEEE, 2021), pp. 1–6
23. S. Murdoch, N. Leaver, Anonymity vs. trust in cyber-security collaboration, in *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security* (2015), pp. 27–29
24. S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. Decentral. Bus. Rev. 21260 (2008)
25. K. Nguyen, S. Pal, Z. Jadidi, A. Dorri, R. Jurdak, A blockchain-enabled incentivised framework for cyber threat intelligence sharing in ICS, in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events* (PerCom BRAINS Workshops) (IEEE, 2022), pp. 261–266
26. L.O. Nweke, S. Wolthusen, Legal issues related to cyber threat information sharing among private entities for critical infrastructure protection, in *2020 12th International Conference on Cyber Conflict (CyCon)*, vol. 1300 (IEEE, 2020), pp. 63–78
27. S. Pal, Extending mobile cloud platforms using opportunistic networks: survey, classification and open issues. J. Univ. Comput. Sci. **21**(12), 1594–1634 (2015)
28. S. Pal, M. Hitchens, T. Rabehaja, S. Mukhopadhyay, Security requirements for the internet of things: a systematic approach. Sensors **20**(20), 5897 (2020)
29. S. Pal, M. Hitchens, V. Varadharajan, On the design of security mechanisms for the internet of things, in *2017 Eleventh International Conference on Sensing Technology (ICST)* (IEEE, 2017), pp. 1–6
30. S. Pal, M. Hitchens, V. Varadharajan, Access control for internet of things-enabled assistive technologies: an architecture, challenges and requirements, in *Assistive Technology for the Elderly* (Elsevier, 2020), pp. 1–43
31. S. Pal, M. Hitchens, V. Varadharajan, T. Rabehaja, Fine-grained access control for smart healthcare systems in the internet of things. EAI Endorsed Trans. Ind. Netw. Intell. Syst. **4**(13) (2018)

32. S. Pal, M. Hitchens, V. Varadharajan, T. Rabehaja, Policy-based access control for constrained healthcare resources, in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (IEEE, 2018), pp. 588–599
33. D. Preuveneers, W. Joosen, J. Bernal Bernabe, A. Skarmeta, Distributed security framework for reliable threat intelligence sharing. Secur. Commun. Netw. **2020** (2020)
34. T. Rabehaja, S. Pal, M. Hitchens, Design and implementation of a secure and flexible access-right delegation for resource constrained environments. Future Gener. Comput. Syst. **99**, 593–608 (2019)
35. R. Riesco, X. Larriva-Novo, V.A. Villagrá, Cybersecurity threat intelligence knowledge exchange based on blockchain. Telecommun. Syst. **73**(2), 259–288 (2020)
36. T. Ring, Threat intelligence: why people don't share. Comput. Fraud Secur. **2014**(3), 5–9 (2014)
37. R. Sams, A note on cryptocurrency stabilisation: seigniorage shares. Brave New Coin 1–8 (2015)
38. C. Sauerwein, C. Sillaber, A. Mussmann, R. Breu, Threat intelligence sharing platforms: an exploratory study of software vendors and research perspectives (2017)
39. T. Schaberreiter, V. Kupfersberger, K. Rantos, A. Spyros, A. Papanikolaou, C. Ilioudis, G. Quirchmayr, A quantitative evaluation of trust in the quality of cyber threat intelligence sources, in *Proceedings of the 14th International Conference on Availability, Reliability and Security* (2019), pp. 1–10
40. B. Shin, P.B. Lowry, A review and theoretical explanation of the 'cyberthreat-intelligence (CTI) capability' that needs to be fostered in information security practitioners and how this can be accomplished. Comput. Secur. **92**, 101761 (2020)
41. M. Swan, *Blockchain: blueprint for a New Economy* (O'Reilly Media, Inc., 2015)
42. W. Tounsi, H. Rais, A survey on technical threat intelligence in the age of sophisticated cyber attacks. Comput. Secur. **72**, 212–233 (2018)
43. T.D. Wagner, K. Mahbub, E. Palomar, A.E. Abdallah, Cyber threat intelligence sharing: survey and research directions. Comput. Secur. **87**, 101589 (2019)
44. M. Wang, M. Duan, J. Zhu, Research on the security criteria of hash functions in the blockchain, in *Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts* (2018), pp. 47–55
45. W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D.I. Kim, A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access **7**, 22328–22370 (2019)
46. Z. Xiaohui, M. Xianghua, A reputation-based approach using consortium blockchain for cyber threat intelligence sharing (2021). arXiv preprint arXiv:2107.06662

# System Identification Methods for Industrial Control Systems

**Mukhtar Hussain, Colin Fidge, Ernest Foo, and Zahra Jadidi**

**Abstract** System identification is a process of creating a mathematical model of a system from its external observations (inputs and outputs). The concept of discovering models from data is trivial in science and engineering fields. The goal of this chapter is to review the recent development in the field of System Identification from the Automatic Control perspective. In the first part of this chapter, we present a classification of design features of Industrial Control Systems (ICSs). Then we review the literature on system identification techniques for creating models of ICSs. The classification of ICSs allows us to identify limitations and unexplored challenges in the literature on system identification techniques.

**Keywords** System identification · Model discovery · Industrial control systems

## 1 Introduction

In the science and engineering fields, mathematical models play an important role in the formal analysis of a system's behaviour such as design optimisation, prediction, verification, and validation of a system's design [95]. Industrial Control Systems (ICSs) can be legacy systems that are subject to changes and upgrades with time but may not be well documented [7, 18, 35]. Therefore, *system identification* techniques play a vital role in creating models from the observation of the evolution in a system's inputs and outputs [9, 54].

M. Hussain (✉) · C. Fidge
School of Computer Science, Queensland University of Technology, Brisbane, Australia
e-mail: m5.hussain@qut.edu.au

C. Fidge
e-mail: c.fidge@qut.edu.au

E. Foo · Z. Jadidi
School of Information and Communication Technology, Griffith University, Brisbane, Australia
e-mail: e.foo@griffith.edu.au

Z. Jadidi
e-mail: z.jadidi@griffith.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors,
Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_2

ICS is a broad term used for a variety of automated control systems from a simple thermostat to a complex manufacturing plant [17]. Thus, modern ICSs have a huge variation in their characteristics and applications. Hence, rich literature can be found on data-driven model techniques can be found.[1] Moreover, the problem of discovering models of ICSs have been studied in many fields including control theory, computer science, system identification, and machine learning [9, 44, 72]. The purpose of this chapter is to conduct a structured literature review on *system identification* techniques applied for ICSs. We classify *system identification* based on the characteristics of ICSs with the aim to identify the research gaps in the literature and help power future researchers in gaining valuable insights.

System identification is an old and mature field with roots going back several decades. Several survey articles were written in the past on the topic of "data-driven model discovery" techniques [9, 21, 22, 27, 32, 61]. These articles review the literature either based on some specific type of ICSs or compare the data-driven model discovery techniques [21, 22]. A set of surveys on continuous dynamic model discovery approaches and their applications in different fields was presented by Ljung [53–56]. Cabasino et al. [9] presented a survey of event-driven model discovery techniques. Lauer and Bloch wrote a book [44] to review hybrid system identification approaches. Gao et al. presented a two-part survey paper [21, 22] on the comparison of different model discovery techniques for fault diagnosis. Our work is different from the previous surveys of *system identification* methods because we review the existing literature on *system identification* from the perspective of ICS's features.

The rest of the chapter is organised as follows. In Sect. 2, we provide a background on the different design functionalities of ICSs. In Sects. 3, 4, 5 and 6, we review the literature on *system identification* methods based on the classification of ICSs presented in Sect. 2. Finally, in Sect. 7, we summarise the literature review with future directions.

## 2 Classification of Industrial Control Systems

Over the years, ICSs have become an essential part of our daily lives such that they can be found in diverse environments ranging from simple automated doors to complex manufacturing plants [93]. There are different classifications of ICSs which can be found in the literature [3, 13, 17]. However, previous classifications of ICSs discussed in the literature [3] is mainly based on designers' perspective which is not effective from the behavioural modelling perspective. For instance, from the model discovery perspective, researchers are interested in the control mechanism instead of the type of controller. In this section, we have highlighted different configurations and design functionalities of ICSs which are essential elements for accurate modelling and model discovery. The following subsections explain our classification of ICSs' features as shown in Fig. 1 in detail.

[1] In this chapter, we use the terms *model discovery* and *system identification* as synonymous

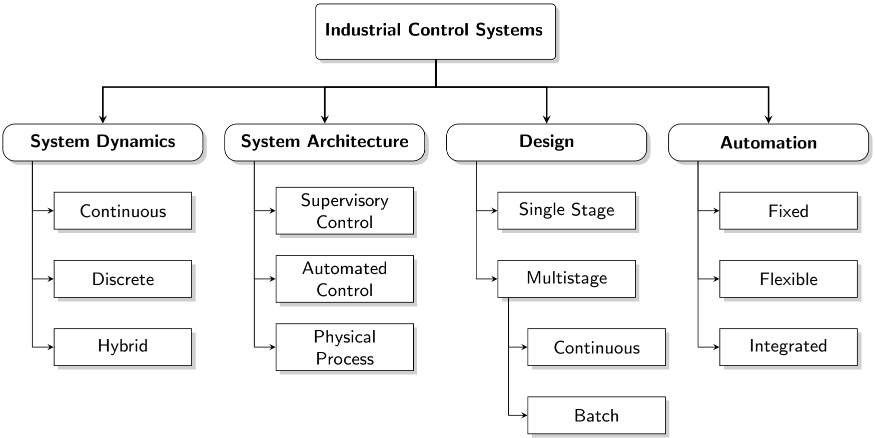


**Fig. 1** A classification of ICSs' feature implementations

## *2.1 System Dynamics*

The second category in our ICS classification framework as shown in Fig. 1 is "system dynamics" which play an important role in the selection of specific modelling and *system identification* approach. The system dynamics can be defined as the evolution of a system's state over time. Based on the type of state evolution, dynamical systems can be classified into three categories:

1. **Continuous Time-driven**, if the state space $X$ is a continuum consisting of $n$-dimensional vectors of real numbers, i.e., it takes values from the Euclidean space $X \in \mathbb{R}^n$ for $n \geq 1$ [10]. Moreover, the system's state evolves continuously with time. Continuous time-driven mainly describe the physical layer process of an ICS. An example of a continuous time-driven dynamic process is a distillation column where the system's state continuously evolves [73].
2. **Discrete Event-driven**, if the state takes values in a countable or finite set $\{q_1, q_2, \ldots, q_N\}$ and states evolve as a sequence of events instead of continuously with time. The event-based state transition mechanism can be seen as simple logical statements of the form "*if something specific happens and the current state is $q_i$, then the next state becomes $q_j$*". The automated and supervisory control layers' process in ICS follows event-driven dynamics. An example of such a system is a simple thermostat, whose state takes on two values, $Q \in \{ON, OFF\}$ where the transition from On mode to another mode is instantaneous. For instance, switching from $ON$ mode to $OFF$ mode and vice versa.
3. **Hybrid**, if the system has heterogeneous (discrete and continuous) dynamics that interact with each other and determine their behaviours over time. Naturally, hybrid dynamics are appropriate to describe the complete operation of most of the ICSs. For instance, a continuous process in a "distillation column" system has

a *start* and *end* which means that the process follows an event-driven transition from the *startup* mode to the *operating* mode and from the *operating* to the *shutdown* mode. Similarly, an event-driven process in a "thermostat" depends on the temperature which evolves continuously. Hybrid dynamics have a central role in ICSs due to the interaction of cyber systems with the physical world. Hybrid dynamics arise in varied systems such as manufacturing, auto pilot design, automotive engine control, traffic control, and chemical processes, among others.

## 2.2 System Architecture

The first category in our ICS classification framework as shown in Fig. 1 is "system architecture". Different design architectures of ICSs can be found in the literature based on the specific application [45, 59]. Our survey in this chapter is devised based on a three-layered architecture of ICSs commonly used for modelling purposes [59] as shown in Fig. 2. The three-layered architecture consists of a supervisory control layer, automatic control layer, and physical layer and captures the main functional features of ICSs. A brief overview of the different layers of the ICS architecture is provided as follows.

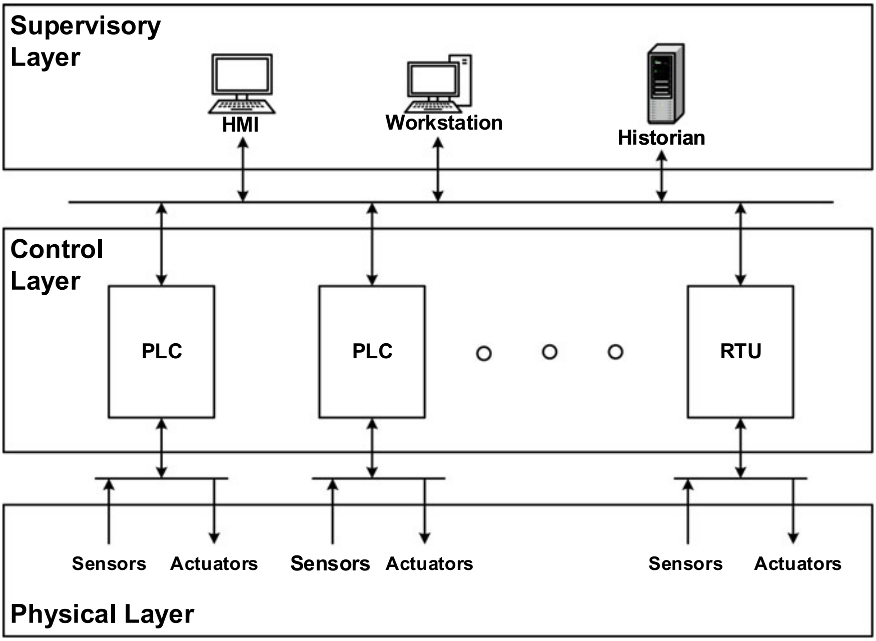


**Fig. 2** Architecture of industrial control systems

1. **Supervisory Control Layer**: The supervisory control layer is responsible for monitoring the operation of the ICS, performing control and supervisory tasks by sending control commands to field controllers, i.e., switching the system's operating modes (e.g., start-up, shutdown, fail-safe) [17]. A decision to switch the system from one operational mode to another is usually based on equipment constraints or working conditions. For example, a nuclear power plant cannot be switched directly to refuelling mode from power generation mode, since switching to refuelling mode must be done through the cold shutdown or the hot shutdown mode. Moreover, switching from the power generation mode to either the cold shutdown or the hot shutdown mode is conditional based on the coolant temperature [41]. The supervisory control level is also monitored by human operators and engineers. Hence, decisions on the priority of tasks to optimise the operational process can also be made [59].
2. **Automated Control Layer**: The automatic control layer regulates the system's physical processes based on the operating mode set by the supervisory control layer. The process is automated using digital controllers such as remote terminal units (RTUs), programmable logic controllers (PLCs), and intelligent electronic devices (IEDs). A digital controller cyclically performs three main steps: (i) input reading, where signals are read from the sensors; (ii) program execution, to determine the new output values for the actuators; and (iii) output writing, where the control signals to the actuators are set. ICSs can be divided broadly into two categories based on the different strategies to automate the physical process by the automated control layer, i.e., process to determine output based on input from sensors.
3. **Physical Process Layer**: The physical layer process represents the continuous evolution of a system's state, e.g., changes in room temperature [17]. The physical process is monitored and controlled using sensors and actuators at the automated control layer. Modelling physical or continuous time-driven behaviour is necessary to forecast, validate and verify a system's behaviour [95].

## *2.3 Design Implementation*

The third category in our classification of ICSs is the design implementation of an ICS. The basic choices for ICS design implementation are:

**Single-stage System** is a combination of physical devices to automate a specific task typically by only one digital controller. Single-stage systems are also referred to as centralised control systems in the process control industry [3]. An example of such a system is an Unmanned Aerial Vehicle (UAV) which is automated by a single controller [37].

**Multistage System** is a combination of single-stage systems cascaded together to automate complex processes. An example of such a system is a water treatment plant [28]. Each unit performs a specific task of the whole process. There can be two different configurations for how each stage interacts with each other, i.e., standalone configuration or shared resources configuration. For standalone configurations, each stage can be treated as an individual system for model discovery [78]. Thus, *system identification* methods for single stage systems can be employed for discovering a model of a multistage system. However, a challenge arises in a multistage stage system with shared resources such that all stages operate in parallel but the processes of adjacent stages are affected by the resource shared between them [4]. Almost all the ICS are programmed to perform iterative tasks. However, not all system processes are sequential. Based on the processing type, ICSs can be described as either continuous processing or batch processing [34, 73] described as follows.

1. **Continuous processing systems** perform a certain set of tasks in order by switching from one mode to another. In a continuous processing system, the whole system is switched from one mode to another. Hence, all state transitions occur and are logged in the sequential timed order. An example of such a system is a nuclear power plant which operates by switching between six plant-wide modes [52]. The operation of a power plant can be described as a continuous processing system because the whole plant can be operating under a certain mode at any certain time instance.
2. **Batch processing systems** are designed to maximize the productivity of the system such as manufacturing systems where multiple units operate concurrently on different batches of a product. An example of such a batch processing system is a water treatment plant [28]. Instead of switching the whole system from one mode to another, each stage is switched to different modes individually based on the batch of product being manufactured [73].

## 2.4 Automation

The last category considered in our classification of an ICS's features which plays an important role in the *system identification* process is the automation strategy in an ICS. The operation category explains the automation strategies and different processing techniques used in ICSs. Industrial processes are time-varying, with non-stationary behaviour, working under diverse operating points named multimode regimes [73]. The ICSs' automation strategies can be divided into three subcategories as follows.

1. **Fixed Automation** is a type of ICS automation in which no human intervention is required to change process parameters or specifications during operations. These types of systems are programmed to do a certain set of tasks *iteratively*. Such systems are deployed in scenarios that once programmed, does not need frequent changes such as elevators, automated doors or traffic signals etc. However, it could be reprogrammed if required [88].
2. **Flexible Automation** is a type of ICS automation in which minimal human intervention is required during the process [88]. Such ICS generally run large-scale

systems through a network of physical devices like actuators or sensors. Human-Machine-Interface (HMI) is designed to provide operators/engineers with a graphical user interface of the whole system, and physical devices can be accessed through it. Operators can change process parameters during operation depending on production requirements. These actions can be either simple turn on or off a device or change the mode of operations. Those user actions are usually logged either with a user ID or station ID [15].

3. **Integrated Automation** is a type of ICS automation in which process operations are human-centric, such as devices like sensors are deployed to facilitate human decision making [88]. An example of such a system is a car manufacturing plant where both human work side-by side with automated plant robots on the assembly line [84]. It is easy to record of inputs and outputs of an automated system. However, the main challenge is to keep a record of every human activity.

To discover an accurate model of an ICS, a model discovery approach should capture the above-mentioned features of ICSs. As mentioned earlier, the model discovery approach is common among many disciplines of science and engineering field, e.g., control theory, system identification, and machine learning [54]. However, the use of different terminologies, as well as the separate reference journals and conferences leads to the fact that very similar solutions to the same problem have been developed independently by different research groups. In the following subsection, we cover the related work on discovering models of ICSs from all these fields (control theory, system identification, and machine learning).

## 3 Model Discovery from the System Dynamics Perspective

System dynamics play an important role in the selection of a model candidate to represent the discovered behaviour [55]. This section provides an overview of system identification techniques from the system dynamics perspective.

### *3.1 Time-Driven Approach*

The behaviour of a continuous time-driven system is an input-output relationship, involving the derivatives (differential equations) or the delayed values (difference equations) of the input-output variables. Let the system's input and output at any time instance $t$ be $x(t)$ and $y(t)$, respectively. A linear difference equation model explaining the relationship between the input and output can be expressed as:

$$y(t+1) + a_0 y(t) + \ldots + a_n y(t-n) = b_0 x(t) + \ldots + b_n x(t-n). \tag{1}$$

Here, we choose to represent the system in discrete time by a difference equation because in ICSs device logs are recorded by sampling (discrete time). Eq. (1) can be expressed in a way to determine the next output value of the system given the previous output and input values as:

$$y(t+1) = -a_0 y(t) - \ldots - a_n y(t-n) + b_0 x(t) + \ldots + b_0 x(t-n). \quad (2)$$

The above equation can be generalised in the form:

$$y(t+1) = f(y(t), x(t)). \quad (3)$$

Once the vectors inputs and outputs are defined, the solution of Eq. (3) can easily be estimated numerically [5]. The literature on time-driven system identification methods is extensive. The methods presented in the literature have been argued based on the model to describe the behaviour of a system and the best way to estimate a time-driven model based on its ability to reproduce output based on input(s) and the presence of noise [55]. For a practical user-oriented introduction, see the MATLAB System Identification Toolbox [62], and there are many other useful references [53, 55, 77]. The main limitation of adopting time-driven system identification methods for discovering a complete model of an ICS is that one equation cannot describe the behaviour of the physical layer process under different operating modes.

## *3.2 Event-Driven Approach*

Similar to the identification of continuous dynamic systems, identification of event-driven systems is determining a mathematical model which describes a relation between inputs and outputs of an event-driven system. An example of an event-driven system is a simple conveyor based package sorting system [66]. The input of the system is recorded using two switch sensors which detect the arrival of a package and its colour. The output event from the controller to sort by changing the direction of a lever (left or right) is based on the input events (package arrival and its colour).

For an event-driven system identification approach, the inputs and outputs should be recorded as a sequence of events instead of sampled data which represents the state of devices at any time instance [4]. Otherwise, it requires data pre-processing to discover an event-driven model as discussed by Estrada-Vargas et al. [86]. Moreover, inputs and outputs of an event-driven system can only take discrete values (e.g., 0 and 1) and any change in the state value of an input or output is considered as an event. The DES identification was first addressed as a problem of discovering finite-state automata [9]. Afterwards, PN models discovery methods were proposed for coping with more complex systems exhibiting concurrent behaviour [86].

Let us provide a brief overview of PNs before discussing DES identification in detail. A PN such as shown in Fig. 3 is a bipartite graph consist of two types of nodes, i.e., places and transitions, and arcs connecting places to transitions and transitions

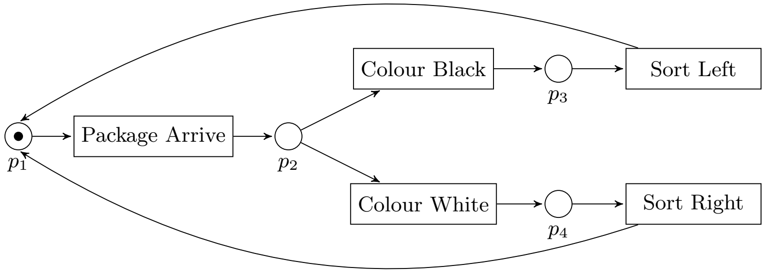


**Fig. 3** An example of a discovered PN model

to places [10]. Conventionally, a place is represented by a circle and a transition is represented by a rectangle. The structure of a PN can be described as a tuple $N = (P, T, A)$, where:

- $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is the finite set of place;
- $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is the finite set of transition;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs from places to transitions and from transitions to places.

The state of a PN is called marking. A marking is a function $m : P \to \mathbb{N}$ that assigns each place a non-negative integer number of tokens. A token is represented by a dot inside a place such as shown in place $p_1$ in Fig. 3. A marked PN imitates the dynamical behaviour of a DES by firing the transitions. A firing of a transition in PN refers to the execution of an event in DES. A transition can only fire if its input place(s) has positive number of token(s). The state of a PN is updated after the firing of a transition by removing a token from the input place(s) of a transition to its output place(s). For instance, the execution of transition (occurrence of the event) "Package Arrive" in the PN model shown in Fig. 3 moves the token from the place $p_1$ to $p_2$.

We explain a DES identification approach using a simple conveyor based package sorting system [66]. The input of the system is recorded using two switch sensors which detect the arrival of a package and its colour. The output event to sort by changing the direction of a lever (left or right) is based on the input events (package arrival and its colour).Let the sequence of the input-output events recorded during the conveyor system's operation be ⟨Package Arrive, Colour Black, Sort Left, Package Arrive, Colour White, Sort Right, Package Arrive, Colour Black, Sort Left⟩. A DES identification approach [86] discovers a PN model such as that shown in Fig. 3 which represents the process identified from the sequence of events. It can be observed that the output command for automated sorting either left or right is conditioned based on the input events, i.e., "package arrival" and its "colour".

Here we should also mention PN models discovery methods from the *process mining* field which is similar to the identification of PN models in the control theory [9]. A few notable differences between DES identification and model discovery techniques from the *process mining* field are as follows. Process mining methods are mainly devoted to discovering PN models of workflow management processes that have a specific *start* and an *end* state. Process mining methods leverage this “start” and “end” information marked in the workflow process logs as “cases”. Therefore, *process mining* methods are inclined towards identifying *workflow nets*, a special class of PN models that must have a “start” and an “end” state. In the DES identification, a system works in a closed-loop, hence, there is no “case” information available in control system logs. Therefore, DES identification is mainly concentrated on “ordinary PN” models. Both ordinary PNs and workflow nets are bounded or 1-safe PN models such that each place can carry only one token at a time [16, 91]. Ordinary PNs are further restricted such that every transition has one incoming arc and one outgoing arc [16]. Therefore, DES identification methods lack in identifying 1-length loops in the PN model [86]. Model discovery methods from the *process mining* field were also adopted for DES identification in the control theory and vice versa [4, 90]. Cabasino et al. [9] presented a comparison between different PN model discovery methods from the process mining and the system identification fields.

## *3.3 Hybrid System Identification*

As mentioned earlier, hybrid dynamic systems are a combination of time-driven and event-driven dynamic subsystems. The investigation of hybrid systems is a fascinating discipline in both control engineering and computer science fields. Hence, hybrid system identification methods can be divided into two groups based on the application area. Researchers from the control engineering field have approached hybrid systems as a collection of differential/difference equations with discontinuous or multi-valued right-hand sides. On the other hand, computer scientists tend to look at hybrid systems primarily as discrete (computer) programs interacting with the physical environment. An overview of *hybrid system identification* methods from both fields is provided in the following subsections.

**Switched Models**: In the control systems and engineering field, *hybrid system identification* is commonly based on Switched AutoRegressive eXogenous (SARX) models which are an extension of time-driven AutoRegressive eXogenous (ARX) models [24]. The mathematical framework of a SARX model can be characterised by a set of continuous modes described by a set of *differential equations* and a logical rule orchestrating switching between the modes [51]. Using the continuous time-driven input-output relation expressed in Eq. 4, the SARX model can be expressed as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t), q(t)), \\ q(t^{+}) &= \delta(x(t), q(t)) \end{aligned} \tag{4}$$

where $x(t) \in \mathbb{R}^m$ represents continuous state vectors, $q$ belongs to a finite set discrete modes $Q = \{q_1, q_2, \ldots, q_N\}$, function $f_q$ characterises the continuous state of the system for the $q$th mode, and $\delta$ explains the discrete switching conditions.

The continuous system identification approaches can be employed to discover the function $f_q$ for each mode for a SARX model [44]. However, it requires the recorded data (input-output device log) to be clustered for each mode. Another challenge in discovering a SARX model is to discover the mode switching conditions [68]. Different approaches have been proposed to address these issues, explained as follows.

The most simple solution to address the issue of mode division and identification of mode switching conditions is mixture modelling [73]. The key idea behind the mixture modelling is to view the identification of multiple ARX models as the identification of a single, 'lifted' ARX model that simultaneously encodes all the ARX submodels. The mixture modelling method can be based on either an algebraic approach [58] or a Bayesian inference approach [94]. The limitation of a mixture modelling approach is it does not incorporate any switching sequence [2].

Another solution based on the identification of mode switching conditions in the control theory literature has been addressed based on piecewise-affine autoregressive exogenous (PWARX) [44]. PWARX models are a subclass of SARX models where the mode invariants divide the input state-space $\mathbb{R}^n$ into polyhedral or mutually exclusive partitions [68]. A piecewise affine (PWA) system can be represented as:

$$\dot{x}(t) = f(x(t), q(t)), \text{ for } x \in \Omega_q \tag{5}$$

with $x(t)$, $y(t)$, $q$, and $f_q$ as in Eq. 4, and $\Omega_q$ divides the state space $\mathbb{R}^n$ such that $\Omega_q \in \mathbb{R}^n$ are disjoint sets. Therefore, the operation regions of PWA systems can be easily identified using clustering methods such as K-means [73]. Hence, mode switching conditions can be identified easily as boundary value limits which is an advantage when trying to construct *guarded* models.

The switched model discovery approaches are able to deal with complex continuous variable dynamics and focus mainly on stability, controllability, robustness and synthesis issues. However, the limitation of SARX and PWARX models is the lack of event-driven structure and non-determinism (the assumption that there is no hidden state), i.e., switching between two modes is only constrained based on the input (sensor) data [23]. However, in many industrial processes switching from one mode to another mode is not always restricted based only on inputs [2]. For example, a nuclear power plant can't be switched directly to *refuelling* mode from *operation* mode, since switching to *refuelling* mode must be done through the *cold shutdown* or *hot shutdown* mode [52]. This discrete state-transition or mode switching information cannot be effectively captured in SARX models [48]. Therefore, SARX and PWARX models are not useful for the system validation and verification which is based on the computation of reachable states for a hybrid system.

Here we should also mention that machine learning approaches such as neural networks, support vector machines, and principal component analysis can also be seen as an extension of time-driven identification methods [44, 56]. However, the limitation of such machine learning approaches is their discovery of black-box models. Hence, these model discovery methods do not provide an insight into the system [18]. Moreover, these models have limited applications for the formal analysis of a control system. For instance, black-box model-based anomaly detection methods have not been effective for diagnosis due to the semantic gap between the model and the system's operation [82].

**Automata Models**: The main limitation of the above-mentioned *hybrid system identification* techniques is that the discovered model are not useful for system validation and verification, i.e., identifying liveness and bottlenecks in the automation process. As mentioned earlier, computer scientists tend to look at hybrid systems primarily as discrete (computer) programs interacting with the physical environment. Moreover, computer scientists are mainly interested in calculating or approximating the reachable state in a model for the safety verification and validation of a hybrid system. Therefore, computer scientists are inclined towards creating finite-state automata (FSA) or Petri net (PN) models

In the computer science field, hybrid system identification methods were first considered by abstracting them as event-driven systems. Most of the work in the computer science field was based on creating an event-driven model (FSA or PN) of hybrid systems using expert knowledge [19, 20, 35, 39]. However, relying solely on expert knowledge for creating large scale ICSs is time consuming and prone to error [4].

Some recent efforts on automatically discovering event-driven models required first converting continuous states into discrete segments to be represented as states in a PN or FSA model [31, 49] as shown in Fig. 4. Here continuous tank level values are converted into four discrete trends such as slow rise (SR), quick rise (QR), quick drop (QD), and stay constant (SC). This segmentation of continuous time-series data is similar to the mode identification discussed in the previous section. However, due to the lack of differentiability in event-driven models to represent the continuous input-output relationships, the segmentation process must be accurate which is challenging [96].

The most common approaches used for the discretisation of continuous signals in the signal processing field are *quantisation* [71] and *piecewise linear representation* [38]. The *quantisation* methods process individual values of the signal instead of trends [47]. On the other hand, there is a tradeoff between "fitness", "precision" and "generalisation" of discrete segmentation using piecewise linear representation [38].

- Precision describes if the piecewise approximation can correctly identify different trends/segments.
- Fitness describes if the piecewise approximation method is resistant to noise and if it can allow some variation in the piecewise approximated segments.
- Generalisation describes if the piecewise linear representation can be used for completely unrelated signals.

Setting an appropriate threshold value in piecewise approximation to differentiate between the different segments is important for automatic classification. Achieving

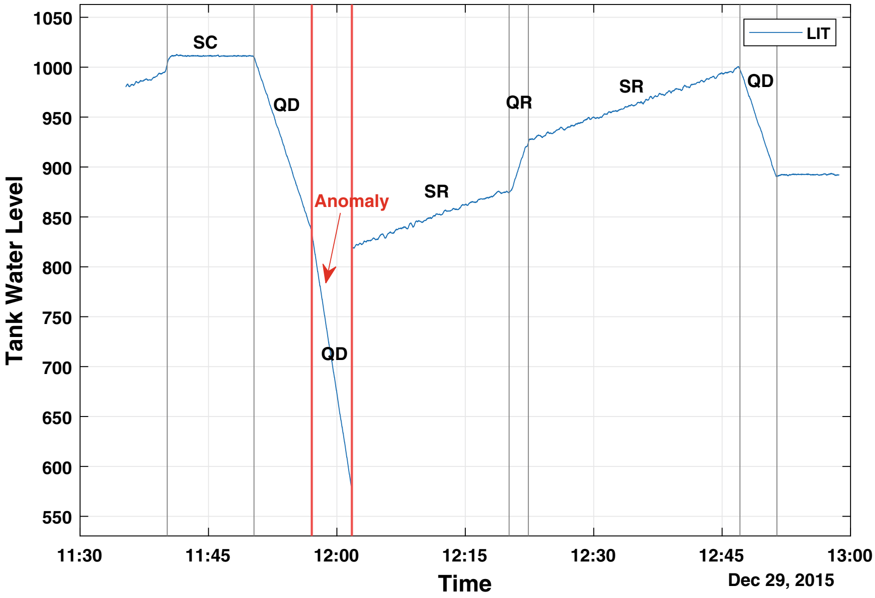


**Fig. 4** Conversion of continuous data to discrete trends

high fitness may lead to an under-fitting approximation. For instance, two slightly different segments are approximated similarly in Fig. 4. On the other hand, achieving high precision may lead to an over-fitting approximation. Hence, it is nearly impossible to classify all the continuous trends accurately without prior knowledge about the system's behaviour or the number of modes. Moreover, the limitation of using an FSA or PN is that these are non-deterministic models. Therefore, the discovered FSA or PN models cannot be used for the prediction of continuous state and stability analysis [48, 96].

**Process Invariant Models**: Another hybrid model discovery approach which is commonly adopted in the computer science field is creating process invariants-based models. Process invariants are mathematical expressions which define the relationship among physical properties in a process automated by digital controller(s) [1]. These methods are mainly designed to learn the automated control layer process of an ICS. The most common approach to determine the automated control process is based on decision tree algorithms [79]. Recently, Mohammadinejad et al. [65] proposed an approach to learn temporal logic expressions to represent discrete state transition conditions based on *decision trees*. However, this approach assumes prior knowledge about the mutually exclusive mode transitions for the decision tree algorithm.

Paul et al. [69] show the development of a physical invariant, based on the theory of Lyapunov-like functions, and a cyber invariant, that governs the correctness of a power dispatch algorithm, and couples the two to develop overall system stability invariant. However, their approach requires prior knowledge, i.e., physics-based models must be known to derive the invariant's parameters. As stated earlier, building accurate process models from expert knowledge is challenging and time consuming [2].

Umer et al. proposed a method to learn control layer design invariants using the Association Rule Learning method [87]. Adepu and Mathur [1] proposed an ICS monitoring method based on system invariants. They employed state entanglement, State Condition Graphs, and state bounds to learn control layer invariants [1]. However, both methods [1, 87] rely on expert knowledge to convert continuous input data into discrete states to discover process invariants.

**Hybrid Automata Models**: The main limitation of the above-mentioned *hybrid system identification* techniques is that they inherit structural restrictions of some sort from the modelling framework being adopted. Therefore, recent publications on *hybrid system identification* approaches in the computer science field are inclined towards combining the discrete event models described by a finite-state automaton and continuous models described by differential or difference equations [23, 42, 76]. The interest is that discovered (hybrid automata or hybrid Petri net) models can be used for a variety of applications such as predictive simulation, verification, stability, and controllability [23, 42, 76]. The benefits of modelling ICSs as "hybrid automata" or "hybrid Petri nets" have been well argued in the literature [11, 26, 29, 74].

To explain the process in detail, let us start with the definition of a hybrid Petri net. A hybrid Petri net can be defined as a tuple $H = (P, T, A, X, Y, G, J, F)$

- $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is a finite set of places;
- $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is a finite set of transitions;
- $A \subseteq (P \times T) \cup (T \times P)$ is the mapping describing the arcs from places to transitions and vice versa;
- $X \subseteq \mathbb{R}^n$ represents the state space where the continuous state variables take values;
- $G : T \rightarrow \mathcal{G}_X$ is a *guard* function that assigns a *guard* $g \in \mathcal{G}_X$ to each transition $t \in T$; and
- $F$ is function that assigns a is a set of differential equations $f$ to each place $p \in P$. The set of differential equations $f_p$ defines the dynamics of a continuous state variable ($x \in X$) against each mode such that $\dot{\mathbf{x}} = f_p(\mathbf{x})$, with cardinality $|F| \leq |P|$.

In this definition, a *guard* $g \in \mathcal{G}_X$ is a logical expression of variables $X$ (e.g., $x > 10$) such that assigning values to variable(s) in the expression, the expression either evaluates to `true` or `false`. The state of a DPN is reflected by tokens, each put in a place $p \in P$. A token can be expressed as a pair $(M, Z)$, where $M$ is a marking function $M : P \rightarrow \mathbb{N}$ for PN $(P, T, A)$ which represents the number of tokens residing inside each place and $Z$ represent values assigned to the state variables $X$. Here, we assumed that a place $p \in P$ in the PN structure can never hold more than one token (1-bounded or safe PN), i.e., $M : P \rightarrow \{0, 1\}$. Moreover, the state of continuous variables $X$ is updated by the differential equations system $F$ at a regular interval (sampling interval).

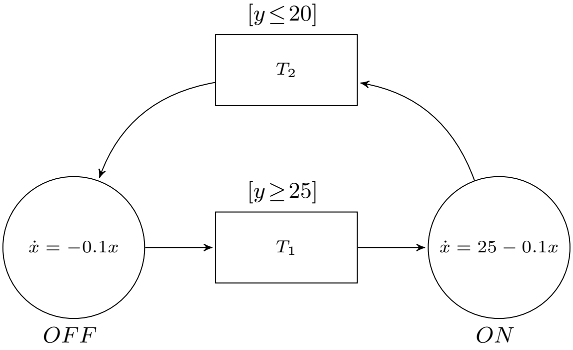


**Fig. 5** An example hybrid Petri net model of a thermostat system

Let us explain *hybrid system identification* techniques with the previously discussed thermostat example. A hybrid Petri net model of the thermostat is shown in Fig. 5. The two discrete states of the thermostat are represented as circles (places of the Petri net model) in Fig. 5. A transition rule is to turn the air-conditioning "on" when the temperature *y* rises above 25 °C, and switch it off when it drops to 20 °C as shown by a PN model in Fig. 5. Meanwhile, the continuous change in the temperature is modelled using differential equations.

The most simple yet effective approach for *hybrid system identification* is to employ existing time-driven and event-driven *system identification* methods [42, 57]. This way we only need to discover the *guards* which combine the time-driven and event-driven dynamic models to complete the hybrid model [11, 29]. The most common *guard* discovery approach used is based on *timed guard* discovery for hybrid automata models [49, 63, 67, 75]. The limitation of this approach is that the discrete state transitions only depend on the *time* feature which is usually not the approach followed in industrial practice [42]. The discovered *guard* conditions can be invalidated if the same system's state is switched after different time intervals which is possible if multiple input variables (sensors) influence a state transition of a system [87].

On the other hand, Soto et al. [23], Blackmore et al. [8], and Balakrishnan et al. [6] proposed *guard* discovery methods for non-deterministic hybrid automata models. In Balakrishnan et al.'s method, *guards* were discovered as probabilistic conditions independent of any system's input or continuous state [6]. Blackmore et al. extended Balakrishnan et al.'s method such that *guard* conditions are based on continuous state variables [8]. The main limitation of their *guard* discovery methods [8, 23] is that *guards* are created as boundary value limits over the states/transitions. These approaches [6, 8, 23] were targeted towards building PWA systems. As discussed earlier, PWA systems form a subclass of ICSs and cannot be generalised for all switched systems. Moreover, their approach discovers non-deterministic models which are not applicable for predictive simulation. In the engineering field, the interest in building deterministic models is for many applications such as real-time anomaly detection and predictive simulation [42, 89].

Lamrani et al. [42], Summerville et al. [85], and Ly and Lipson [57] share the identical motivation, i.e., discovering *guards* for creating deterministic hybrid automata models. Ly and Lipson's approach [57] is based on the symbolic regression method to discover the *guard* conditions as algebraic expressions of input variables which are suitable for describing the behaviour of non-linear systems. However, in practice, *guard* conditions are disjunctions and conjunctions of logical expressions [87]. CHARDA [85] requires prior knowledge of the system to infer *guard* conditions. The limitation of Lamrani et al.'s method is it can only discover the conjunction of condition expressions. Therefore, the *guards* do not reflect an accurate system behaviour in a situation when state transitions are influenced by the disjunction of conditions [87]. Hence discovering accurate *guards* for high fidelity hybrid models is still an open challenge.

**Mode Identification**: A mode can be defined as a certain continuous trajectory/operation of a system [73]. The mode identification is a process of labelling or identifying the system's operating modes information in the recorded data for hybrid mode discovery. Hence, mode identification is the first step of the hybrid model discovery algorithms. Moreover, accurate mode identification is essential for discovering an accurate model of a hybrid system. Identifying modes in an *on-off* switching system [31] or a single output system [23, 63] is relatively straightforward. However, the challenge is to correctly identify modes if modes are indicated by multiple process variables and in the presence of noise.

Hybrid automata model discovery methods are mainly based on rather simple techniques for mode identification. For instance, Saez et al. [76] and Ly and Lipson [57] assume that mode information is available based on expert knowledge. Lin et al. [49] and Soto et al. [23] employed piecewise linear functions to divide a time series into discrete "trends" such that each trend can be represented as a system mode. The limitation of their approach is that piecewise linear approximation introduces a tradeoff between accuracy and tractability [42]. Research on monitoring of multimode process systems using machine learning approaches has made significant progress on mode identification. A brief overview of mode identification approaches from machine learning is provided as follows.

1. **Clustering** methods are unsupervised learning tools used for dividing a data set into various groups or clusters so those observations belonging to the same group are similar and different from the other observations of the data set. The most common approaches used for clustering are as follows.

   – **K-means**: The K-means clustering methods are a popular mode identification choice because of efficiency and easy implementation. Few extensions of the basic K-means clustering method have also been proposed for accurate mode identification. The main obstacle in the application of K-means methods is that the number of clusters to characterise the data must be fixed or known in advance. Moreover, the identification of transient clusters is challenging using K-means methods because the identification of clusters with asymmetric size is a difficult and sometimes even an impossible task [73].

  - **Fuzzy C-means**: FFuzzy C-means allows the soft assignment of the observation, unlike K-means clustering methods where data observations is clustered based on hard threshold values. In fuzzy clustering, an observation can belong to multiple cluster such that their membership is based on data trend [92]. Different extensions of fuzzy C-means such as Kernel Fuzzy C-means and distance-based Fuzzy C-means have been proposed for clustering multimode processes data [73]. The main advantage of clustering-based mode identification approaches is the capability of dealing with outliers or data from nonlinear processes. However, the main weakness of the proposals made is that their application is limited to data sets formed only by steady modes [73].

2. **Window-based approaches**. The data from industrial processes are time-series. Thus, it's logical to have developed methods that sequentially identify clusters by following the time direction [38]. Window-based methods use a moving window subset of the series, and similarity is measured by considering the spatial and temporal information of the features. The most common approach used in many papers is based on changes of mean and variance in the distribution as an indicator of a mode change [12, 83]. However, when the distribution of the modes contains transition intervals/modes, different features should be analysed. For instance, density-based definitions of clusters have been modified for developing mode identification algorithms [81]. The main advantage of window-based approaches is that the number of modes is determined automatically, and most of them can identify transitions from steady modes. However, implementing moving window methods is difficult because selecting a similarity measure and designing a strategy for distinguishing transitions from steady modes are not easy tasks because one approach cannot be generalised for all the systems [73]. Hence, accurate identifying modes from the dataset is still an open challenge.

## 4 Model Discovery from the System Architecture Perspective

This section provides an overview of model discovery literature from the perspective of ICS architecture as discussed in Sect. 2.2. The model discovery methods discussed in the previous section capture the behaviour of an ICS from a certain level of abstraction. For instance, time-driven system identification methods are suitable if someone is interested in discovering a model of the physical layer process [27, 55]. Meanwhile, event-driven system identification methods are suitable for discovering models of the automated and supervisory control layer processes. Moreover, hybrid system identification methods [63, 76, 85] mainly concentrate on modelling the combined behaviour of automated control layer and physical layer processes.

Most ICSs, such as automated highway systems, air traffic management systems, and unmanned aerial vehicles, are multi-objective systems. In those ICSs, many tasks are accomplished by a supervisory control layer by appropriately enabling/disabling low-level automated control process modes [25]. A hierarchical model which incorporates an ICS's system architecture perspective can be created based on a hierarchical modelling framework [40]. The concept of creating a hierarchical model of ICSs is not new in the control system field [25, 33, 64]. However, hierarchical modelling approaches mainly rely solely on expert knowledge to create an ICS model [36, 40, 50]. As stated earlier, relying solely on expert knowledge is time consuming and prone to errors.

## 5 Model Discovery from the Design Perspective

From the design perspective, discovering a single-stage ICS is rather simple. However, challenges arise for discovering a model of a multistage ICS. The first challenge arises based on the interactions between the different stages of a multistage ICS. If there is no shared resource between different stages than each stage can be modelled as a standalone system [78]. This approach is presented by Saives et al. [78] for discovering a model of multistage discrete event-driven system. Saives et al.'s method [78] can also be applied for creating a model of a multistage hybrid system. However, a limitation of Saives et al.'s method [78] is that it is not applicable on a multistage ICS with shared resources. In this case, it is necessary to identify a model of the whole system instead of treating each stage as a standalone system to identify deadlocks or faults caused by shared resources [4]. Allen and Tilbury [4] proposed a solution for discovering a model of a multistage event-driven system with shared resources. However, creating a model of multistage hybrid system with shared resources is still an open challenge. Existing hybrid system identification approaches [23, 44, 63] are applicable on a single stage hybrid systems.

Another challenge for modelling a multistage ICS is based on the prouction policies. In most publications, system identification methods consider a continuous processing perspective for model discovery which involves creating a monolithic model of all stages of ICSs. Capturing the batch processing perspective of a multistage ICS in the models adds an interesting challenge [73]. The reason is that there are two different points of view with regard to how the multimodal condition manifests in a multistage batch processing ICS [73]. If production policies or environmental conditions vary with the batch only, the multimode feature can be reflected as a batch-to-batch variation. On the other hand, if the batch processes undergo different stages, each stage can be considered as an operating condition and the batch process itself could then be considered as a multimode process. The simultaneous analysis of both features becomes a challenging problem address.

## 6 Model Discovery from the Automation Perspective

Capturing the automation perspective of an ICS by a model discovery approach is one of the most challenging tasks. In this section, we highlight the challenges in creating dynamic models from the automation perspective.

Discovering a time-driven model of the physical process of an ICS has minimum or no impact from the automation perspective. Let's consider a water tank system, to discover a time-driven model of the water level in a tank as a function of flow-in and flow-out of the water tank, time-driven model discovery techniques only require inputs and outputs to estimate the model [55]. It does not matter that the flow-in and flow-out valves are opened and closed either automatically or manually.

However, the challenge arises while discovering event-driven or hybrid models of an ICS. Most of the work presented in the literature on both types of methods considers fixed automation processes [9, 44]. A reason for that is discovering a model of flexible automation and integrated automation based ICSs is to identify human activity. Unlike computer control mechanisms where decisions are made based on a pre-determined set of rules, human information processing and decision making is based on three cognitive levels: rule-based, knowledge-based, and skill-based [80].

There has been interest in human activity recognition to identify anomalies and consequences of human decisions in an ICS [46, 70]. Human activity recognition is a two-step process, i.e., data collection and feature extraction [43]. As mentioned in the introduction that "system identification" or "model discovery" is a process of creating models from the data when there is very little or no information about the system's working is available. The availability of data is a crucial prerequisite for a model discovery process. The automation process highly influences the availability of data. Moreover, the result of any model discovery approach can be no better than what corresponds to the information contents in the data [55].

In flexible automation systems, human activity is limited in a sense that operators and engineers interact with the system through Human Machine Interface (HMI) at the supervisory control layer [14]. Hence, human activity can easily be recorded in the ICSs' device logs [15]. However, learning a model of human decisions from the recorded data set is a challenging problem to address. In discrete event-driven systems, human activity can be limited to discrete events. Hence, it can be seen as supervisory control of discrete event systems where human activity can be modelled as human-controlled events [46]. On the other hand, in a hybrid dynamic system, inferring human decisions from the dataset is an open challenge. Recently, Hussain et al. [30] proposed a method to discover an operator's decision making at the supervisory control layer of flexible automation-based hybrid systems. Their approach can only identify linear trend-based decisions, for instance, turn-off heater because the temperature of the furnace was rising very quickly. However, it may not be applicable for discovering human skills-based complicated decisions.

Human-activity recognition in an integrated automation scenario where human and automated systems work side by side pose a different kind of a challenge. The main issue in discovering a model of an integrated-automation system is the availability of data. As mentioned earlier that the system identification process starts with the data and the results of a model discovery process can be no better than the information contents in the data. Stiefmeier et al. [84] and Mannhardt et al. [60] proposed using wearable sensors and human activity recognition to generate and process the data for system identification approaches. Stiefmeier et al.'s method creates an event-driven model. Hence, it is applicable for identifying faults such as a worker missed a step in the process [84]. A possible extension of their work can be in the hybrid systems' field to identify if the worker has completed the task accurately.

## 7 Conclusion

The large scale use of ICSs in critical infrastructures raises concerns for the safety and reliability of critical infrastructure. The safety and reliability of ICSs can be ensured using simulation and model checking tools. However, the essential element for model-based engineering, i.e., an accurate model of a system, is not always available. In previous sections, we provided an overview of model discovery methods from the perspective of different design features of ICSs. We have identified some challenges in the system identification literature as follows.

- Several hybrid model discovery approaches can be found in the literature. The most effective approach to model a hybrid system is as a hybrid automata such that finite-state automaton is used to describe the event-driven dynamics and the discrete part of the hybrid state is represented by the net marking. The continuous part of the state is described by additional variables, whose dynamics is ruled by differential algebraic equations (DAE) not represented in the net structure. The main challenge in discovering accurate hybrid automata is to identify accurate *guards* that describe when discrete state transitions may occur based on the continuous state evolution.
- Existing *hybrid system identification* approaches for creating models of ICSs implicitly assume that all mode switching activities in the ICS are on the same abstraction level, i.e., automatic control layer. However, this is not always the case with flexible automation systems. As mentioned earlier, unlike the automated control process, the supervisory control level process is monitored by human operators and engineers. Therefore, decisions at the supervisory control level of an ICS varies which can be based on operator's experience, skill, or pre-determined rules. A model discovery approach which incorporates an operator's actions at supervisory control level is still an open challenge.
- Existing literature on *hybrid system identification* methods is mainly concentrated on a single-stage ICS. These methods can be extended for a multistage system such that each stage operates in a standalone setting. However, problems arise for multiple stage systems with shared resources. Allen and Tilbury [4] proposed a solution to address this issue for event-driven systems. Nevertheless, discovering a model of a multistage hybrid system with shared resources is still an open challenge.

## References

1. S. Adepu, A. Mathur, Distributed attack detection in a water treatment plant: method and case study. IEEE Trans. Dependable Secure Comput. **18**(1), 86–99 (2021). https://doi.org/10.1109/TDSC.2018.2875008
2. M.S. Afzal, W. Tan, T. Chen, Process monitoring for multimodal processes with mode-reachability constraints. IEEE Trans. Ind. Electron. **64**(5), 4325–4335 (2017). https://doi.org/10.1109/TIE.2017.2677351
3. J. Agre, L. Clare, S. Sastry, A taxonomy of distributed real-time control systems, in *Advances in Computers*, vol. 49, ed. by M. Zelkowitz (Elsevier Science & Technology, 1999), pp. 303–352. https://doi.org/10.1016/S0065-2458(08)60288-0. https://linkinghub.elsevier.com/retrieve/pii/S0065245808602880
4. L.V. Allen, D.M. Tilbury, Anomaly detection using model generation for event-based systems without a preexisting formal model. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. **42**(3), 654–668 (2012). https://doi.org/10.1109/TSMCA.2011.2170418
5. K.J. Åström, B. Torsten, Numerical identification of linear dynamic systems from normal operating records. IFAC Proc. Volumes **2**(2), 96–111 (1965). https://doi.org/10.1016/s1474-6670(17)69024-4
6. H. Balakrishnan, I. Hwang, J.S. Jang, C.J. Tomlin, Inference methods for autonomous stochastic linear hybrid systems, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2993 (Springer, Berlin, Heidelberg, 2004), pp. 64–79. https://doi.org/10.1007/978-3-540-24743-2_5, http://link.springer.com/10.1007/978-3-540-24743-2_5
7. M. Biro, A. Mashkoor, J. Sametinger, R. Seker, Software safety and security risk mitigation in cyber-physical systems. IEEE Softw. **35**(1), 24–29 (2017). https://doi.org/10.1109/MS.2017.4541050
8. L. Blackmore, S. Gil, S. Chung, B. Williams, Model learning for switching linear systems with autonomous mode transitions, in *2007 46th IEEE Conference on Decision and Control* (IEEE, 2007), pp. 4648–4655. https://doi.org/10.1109/CDC.2007.4434779, http://ieeexplore.ieee.org/document/4434779/
9. M.P. Cabasino, P. Darondeau, M.P. Fanti, C. Seatzu, Model identification and synthesis of discrete-event systems, in *Contemporary Issues in Systems Science and Engineering* (Wiley, Hoboken, NJ, USA, 2015), pp. 343–366. https://doi.org/10.1002/9781119036821.ch10, http://doi.wiley.com/10.1002/9781119036821.ch10
10. C.G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, 2nd edn. (Springer US, Boston, MA, 2008). https://doi.org/10.1007/978-0-387-68612-7_1
11. R. Champagnat, P. Esteban, H. Pingaud, R. Valette, Modeling hybrid systems by means of high-level petri nets: benefits and limitations. IFAC Proc. Volumes **30**(6), 349–354 (1997). https://doi.org/10.1016/s1474-6670(17)43389-1, http://dx.doi.org/10.1016/S1474-6670(17)43389-1
12. Y. Chang, R. Ma, F. Wang, W. Zheng, S. Wang, Multimode process mode identification with coexistence of quantitative information and qualitative information. IEEE Trans. Autom. Sci. Eng. 1–12 (2020). https://doi.org/10.1109/tase.2019.2963550
13. M.H. Cintuglu, O.A. Mohammed, K. Akkaya, A.S. Uluagac, A survey on smart grid cyber-physical system testbeds. IEEE Commun. Surv. Tutor. **19**(1), 446–464 (2017). https://doi.org/10.1109/COMST.2016.2627399
14. T. Cucinotta, A. Mancina, G.F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo, F. Rusinà, A real-time service-oriented architecture for industrial automation. IEEE Trans. Ind. Inform. **5**(3), 267–277 (2009). https://doi.org/10.1109/TII.2009.2027013
15. A. Daneels, W. Salter, WHAT IS SCADA? in *International Conference on Accelerator and Large Experimental Physics Control Systems* (Trieste, Italy, 1999), pp. 339–343
16. R. David, H. Alla, Petri nets for modeling of dynamic systems. A survey. Automatica **30**(2), 175–202 (1994). https://doi.org/10.1016/0005-1098(94)90024-8

17. V.L. Do, L. Fillatre, I. Nikiforov, P. Willett, Feature article: security of SCADA systems against cyber–physical attacks. IEEE Aerosp. Electron. Syst. Mag. **32**(5), 28–45 (2017). https://doi.org/10.1109/MAES.2017.160047, http://ieeexplore.ieee.org/document/7954148/
18. S. Etalle, From intrusion detection to software design, in *European Symposium on Research in Computer Security* (2017), pp. 1–10. https://doi.org/10.1007/978-3-319-66402-6_1, http://link.springer.com/10.1007/978-3-319-66399-9http://link.springer.com/10.1007/978-3-319-66402-6_1
19. S. Faltinski, H. Flatt, F. Pethig, B. Kroll, A. Vodenčarević, A. Maier, O. Niggemann, Detecting anomalous energy consumptions in distributed manufacturing systems, in *IEEE International Conference on Industrial Informatics (INDIN)* (2012), pp. 358–363. https://doi.org/10.1109/INDIN.2012.6301142
20. D. Fauri, D.R. Dos Santos, E. Costante, J. Den Hartog, S. Etalle, S. Tonetta, From system specification to anomaly detection (and back), in *Workshop on Cyber-Physical Systems Security and Privacy* (2017), pp. 13–24. https://doi.org/10.1145/3140241.3140250, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85037147831&doi=10.1145
21. Z. Gao, C. Cecati, S. Ding, A survey of fault diagnosis and fault-tolerant techniques part II: fault diagnosis with knowledge-based and hybrid/active approaches. IEEE Trans. Ind. Electron. **62**(6), 1–1 (2015). https://doi.org/10.1109/TIE.2015.2419013, http://ieeexplore.ieee.org/document/7076586/
22. Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques—Part I: fault diagnosis with model-based and signal-based approaches. IEEE Trans. Ind. Electron. **62**(6), 3757–3767 (2015). https://doi.org/10.1109/TIE.2015.2417501, http://ieeexplore.ieee.org/document/7069265/
23. M. García Soto, T.A. Henzinger, C. Schilling, L. Zeleznik, Membership-based synthesis of linear hybrid automata. Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11561 LNCS**(754411), 297–314 (2019). https://doi.org/10.1007/978-3-030-25540-4_16
24. A. Garulli, S. Paoletti, A. Vicino, A survey on switched and piecewise affine system identification. IFAC Proc. Volumes **45**(16), 344–355 (2012). https://doi.org/10.3182/20120711-3-BE-2027.00332. https://linkinghub.elsevier.com/retrieve/pii/S1474667015379751
25. B. Gaudin, H. Marchand, Supervisory control of product and hierarchical discrete event systems. Euro. J. Control **10**(2), 131–145 (2004). https://doi.org/10.3166/ejc.10.131-145
26. L. Ghomri, H. Alla, Modeling and analysis using hybrid Petri nets. Nonlinear Anal.: Hybrid Syst. **1**(2), 141–153 (2007). https://doi.org/10.1016/j.nahs.2006.04.004
27. J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N.O. Tippenhauer, H. Sandberg, R. Candell, A survey of physics-based attack detection in cyber-physical systems. ACM Comput. Surv. **51**(4), 1–36 (2018). https://doi.org/10.1145/3203245, http://dl.acm.org/citation.cfm?doid=3236632.3203245
28. J. Goh, S. Adepu, K.N. Junejo, A. Mathur, A dataset to support research in the design of secure water treatment systems, in *International Conference on Critical Information Infrastructures Security* (2017), pp. 88–99. https://doi.org/10.1007/978-3-319-71368-7_8, http://link.springer.com/10.1007/978-3-319-71368-7_8
29. T.A. Henzinger, The theory of hybrid automata, in *Verification of Digital and Hybrid Systems* (Springer, Berlin, Heidelberg, 2000), pp. 265–292. https://doi.org/10.1007/978-3-642-59615-5_13, http://link.springer.com/10.1007/978-3-642-59615-5_13
30. M. Hussain, C. Fidge, E. Foo, Z. Jadidi, Discovering data-aware mode-switching constraints to monitor mode-switching decisions in supervisory control. IEEE Trans. Ind. Inform. **18**(6), 3734–3743 (6 2022). https://doi.org/10.1109/TII.2021.3120020, https://ieeexplore.ieee.org/document/9573395/
31. M. Hussain, E. Foo, S. Suriadi, An improved industrial control system device logs processing method for process-based anomaly detection, in *International Conference on Frontiers of Information Technology (FIT)* (IEEE, 2019), pp. 150–1505. https://doi.org/10.1109/FIT47737.2019.00037, https://ieeexplore.ieee.org/document/8991656/

32. R. Isermann, Model-based fault-detection and diagnosis—Status and applications. Ann. Rev. Control **29**(1), 71–85 (2005). https://doi.org/10.1016/j.arcontrol.2004.12.002
33. K. Jensen, Coloured petri nets: a high level language for system design and analysis, in *High-level Petri Nets* (Springer, Berlin, Heidelberg, 1991), pp. 342–416. https://doi.org/10.1007/3-540-53863-1_31, http://link.springer.com/10.1007/3-540-53863-1_31
34. Q. Jiang, S. Yan, X. Yan, H. Yi, F. Gao, Data-driven two-dimensional deep correlated representation learning for nonlinear batch process monitoring. IEEE Trans. Ind. Inform. **16**(4), 2839–2848 (2020). https://doi.org/10.1109/TII.2019.2952931
35. X. Jin, A. Donze, J.V. Deshmukh, S.A. Seshia, Mining requirements from closed-loop control models. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **34**(11), 1704–1717 (11 2015). https://doi.org/10.1109/TCAD.2015.2421907, http://ieeexplore.ieee.org/document/7084172/
36. K. Kang, L. Xu, W. Wang, G. Wu, J. Wei, W. Shi, J. Li, A hierarchical automata based approach for anomaly detection in smart home devices, in *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)* (IEEE, 2020), pp. 1–8. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00021. https://ieeexplore.ieee.org/document/9291572/
37. A. Keipour, M. Mousaei, S. Scherer, ALFA: a dataset for UAV fault and anomaly detection. Int. J. Robot. Res. 027836492096664 (2020). https://doi.org/10.1177/0278364920966642. http://journals.sagepub.com/doi/10.1177/0278364920966642
38. E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in *Proceedings 2001 IEEE International Conference on Data Mining* IEEE Comput. Soc (2001), pp. 289–296. https://doi.org/10.1109/ICDM.2001.989531. http://ieeexplore.ieee.org/document/989531/
39. Z. Kong, A. Jones, C. Belta, Temporal logics for learning and detection of anomalous behavior. IEEE Trans. Autom. Control **62**(3), 1210–1222 (2017). https://doi.org/10.1109/TAC.2016.2585083
40. T.J. Koo, G.J. Pappas, S. Sastry, Mode switching synthesis for reachability specifications, in *Hybrid Systems: Computation and Control* (Springer, Berlin, Heidelberg, 2001), pp. 333–346. https://doi.org/10.1007/3-540-45351-2_28. http://link.springer.com/10.1007/3-540-45351-2_28
41. X.D. Koutsoukos, P.J. Antsaklts, J.A. Stiver, M.D. Lemmon, Supervisory control of hybrid systems. Proc. IEEE **88**(7), 1026–1049 (2000). https://doi.org/10.1109/5.871307
42. I. Lamrani, A. Banerjee, S.K.S. Gupta, HyMn: mining linear hybrid automata from input output traces of cyber-physical systems, in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)* (IEEE, 2018), pp. 264–269. https://doi.org/10.1109/ICPHYS.2018.8387670. https://ieeexplore.ieee.org/document/8387670/
43. O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**(3), 1192–1209 (2013). https://doi.org/10.1109/SURV.2012.110112.00192. http://ieeexplore.ieee.org/document/6365160/
44. F. Lauer, G. Bloch, Hybrid system identification, lecture notes in control and information sciences, vol. 478 (Springer International Publishing, 2019). https://doi.org/10.1007/978-3-030-00193-3, http://link.springer.com/10.1007/978-3-030-00193-3
45. J. Lee, B. Bagheri, H.A. Kao, A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf. Lett. **3**, 18–23 (2015). https://doi.org/10.1016/j.mfglet.2014.12.001
46. J.S. Lee, M.C. Zhou, P.L. Hsu, An application of Petri nets to supervisory control for human-Computer interactive systems. IEEE Trans. Ind. Electron. **52**(5), 1220–1226 (2005). https://doi.org/10.1109/TIE.2005.855694
47. D. Liberzon, *Switching in Systems and Control, Systems & Control: Foundations & Applications*, vol. 53 (Birkhäuser Boston, Boston, MA, 2003). https://doi.org/10.1007/978-1-4612-0017-8. http://link.springer.com/10.1007/978-1-4612-0017-8

48. H. Lin, P.J. Antsaklis, Hybrid dynamical systems: an introduction to control and verification, vol. 1 (Now Foundations and Trends, 2014). https://doi.org/10.1561/2600000001, http://www.nowpublishers.com/articles/foundations-and-trends-in-systems-and-control/SYS-001
49. Q. Lin, S. Adepu, S. Verwer, A. Mathur, TABOR: a graphical model-based approach for anomaly detection in industrial control systems, in *ASIA CCS—ACM Asia Conference on Computer and Communications Security*, vol. 12 (2018), 525–536 (2018). https://doi.org/10.1145/3196494.3196546. http://dl.acm.org/citation.cfm?doid=3196494.3196546
50. S. Liu, X. Hu, J. Wang, Hierarchical modeling fault-error-failure dependencies for cyber-physical systems, in *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013, Advances in Intelligent Systems and Computing*, vol. 212 (Springer, Berlin, Heidelberg, 2013), pp. 641–649. https://doi.org/10.1007/978-3-642-37502-6_77, http://link.springer.com/10.1007/978-3-642-37502-6_77
51. X. Liu, P. Stechlinski, Hybrid and switched systems, in *Infectious Disease Modeling. Nonlinear Systems and Complexity, Nonlinear Systems and Complexity*, vol. 19, chap. 2 (Springer International Publishing, 2017), pp. 21–39. https://doi.org/10.1007/978-3-319-53208-0_2, http://link.springer.com/10.1007/978-3-319-53208-0
52. J. Livingston, The nuclear electrical engineer, an educational resource for electrical engineers in the nuclear power industry (2014). http://www.nuclearelectricalengineer.com/nuclear-power-plant-modes-explained-here/. Accessed 31 Jan. 2020
53. L. Ljung, Experiments with identification of continuous time models, vol. 42 (IFAC, 2009). https://doi.org/10.3182/20090706-3-fr-2004.00195
54. L. Ljung, Perspectives on system identification. Ann. Rev. Control **34**(1), 1–12 (4 2010). https://doi.org/10.1016/j.arcontrol.2009.12.001. https://linkinghub.elsevier.com/retrieve/pii/S1367578810000027
55. L. Ljung, System identification, in *Wiley Encyclopedia of Electrical and Electronics Engineering* (Wiley, Hoboken, NJ, USA, 2017), pp. 1–19. https://doi.org/10.1002/047134608X.W1046.pub2. http://doi.wiley.com/10.1002/047134608X.W1046.pub2
56. L. Ljung, H. Hjalmarsson, H. Ohlsson, Four encounters with system identification. Euro. J. Control **17**(5-6), 449–471 (2011). https://doi.org/10.3166/ejc.17.449-471
57. D.L. Ly, H. Lipson, Learning symbolic representations of hybrid dynamical systems. J. Mach. Learn. Res. **13**, 3585–3618 (2012)
58. Y. Ma, R. Vidai, Identification of deterministic switched ARX systems via identification of algebraic varieties. Lect. Notes Comput. Sci. **3414**, 449–465 (2005). https://doi.org/10.1007/978-3-540-31954-2_29
59. N. Mahdavi Tabatabaei, S. Najafi Ravadanegh, N. Bizon, (eds.), Power systems resilience, in *Power Systems* (Springer International Publishing, Cham, 2019). https://doi.org/10.1007/978-3-319-94442-5. http://search.ebscohost.com/login.aspx?direct=true&db=edsebk&AN=1875752&site=eds-live. http://link.springer.com/10.1007/978-3-319-94442-5
60. F. Mannhardt, R. Bovo, M.F. Oliveira, S. Julier, A taxonomy for combining activity recognition and process discovery in industrial environments, in *International Conference on Intelligent Data Engineering and Automated Learning*, vol. 8206 (Springer International Publishing, 2018), pp. 84–93. https://doi.org/10.1007/978-3-030-03496-2_10, http://link.springer.com/10.1007/978-3-642-41278-3. http://link.springer.com/10.1007/978-3-030-03496-2_10
61. J. Marzat, H. Piet-Lahanier, F. Damongeot, E. Walter, Model-based fault diagnosis for aerospace systems: a survey, in Proc. Instit. Mech. Eng. Part G: J. Aerosp. Eng. **226**(10), 1329–1360 (10 2012). https://doi.org/10.1177/0954410011421717. http://journals.sagepub.com/doi/10.1177/0954410011421717
62. System Identification Toolbox. https://au.mathworks.com/products/sysid.html. Accessed 01 Aug. 2021
63. R. Medhat, S. Ramesh, B. Bonakdarpour, S. Fischmeister, A framework for mining hybrid automata from input/output traces, in *2015 International Conference on Embedded Software (EMSOFT)* (IEEE, 2015), pp. 177–186. https://doi.org/10.1109/EMSOFT.2015.7318273. http://ieeexplore.ieee.org/document/7318273/

64. E. Mikk, Y. Lakhnechi, M. Siegel, Hierarchical automata as model for statecharts, in *Advances in Computing Science—ASIAN'97. ASIAN 1997. Lecture Notes in Computer Science*, vol. 1345 (Springer, Berlin, Heidelberg, 1997), pp. 181–196. https://doi.org/10.1007/3-540-63875-X_52. http://link.springer.com/10.1007/3-540-63875-X_52
65. S. Mohammadinejad, J.V. Deshmukh, A.G. Puranic, Mining environment assumptions for cyber-physical system models, in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)* (IEEE, 2020), pp. 87–97. https://doi.org/10.1109/ICCPS48487.2020.00016. https://ieeexplore.ieee.org/document/9096037/
66. D. Myers, S. Suriadi, K. Radke, E. Foo, Anomaly detection for industrial control systems using process mining. Comput. Secur. **78**, 103–125 (2018). https://doi.org/10.1016/j.cose.2018.06.002
67. O. Niggemann, B. Stein, A. Maier, A. Vodenčarević, H.K. Büning, Learning behavior models for hybrid timed systems, in *Proceedings of the National Conference on Artificial Intelligence*, vol. 2 (2012), pp. 1083–1090
68. S. Paoletti, A.L. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems a tutorial. Euro. J. Control **13**(2–3), 242–260 (2007). https://doi.org/10.3166/ejc.13.242-260
69. T. Paul, J.W. Kimball, M. Zawodniok, T.P. Roth, B. McMillin, Invariants as a unified knowledge model for cyber-physical systems, in *2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)* (IEEE, 2011), pp. 1–8. https://doi.org/10.1109/SOCA.2011.6166223. http://ieeexplore.ieee.org/document/6166223/
70. E. Pricop, J. Fattahi, N. Dutta, M. Ibrahim, (eds.), Recent developments on industrial control systems resilience, studies in systems, decision and control, vol. 255 (Springer International Publishing, Cham, 2020). https://doi.org/10.1007/978-3-030-31328-9. http://link.springer.com/10.1007/978-3-030-31328-9
71. J.G. Proakis, D.G. Manolakis, *Digital Signal Processing: principles, Algorithms, and Applications*, 3rd edn. (Prentice-Hall of Australia Pty. Limited, Sydney Prentice-Hall, 1996). https://engineering.purdue.edu/~ee538/DSP_Text_3rdEdition.pdf
72. M. Quinones-Grueiro, A. Prieto-Moreno, O. Llanes-Santiago, Modeling and monitoring for transitions based on local kernel density estimation and process pattern construction. Ind. Eng. Chem. Res. **55**(3), 692–702 (2016). https://doi.org/10.1021/acs.iecr.5b03902
73. M. Quiñones-Grueiro, A. Prieto-Moreno, C. Verde, O. Llanes-Santiago, Data-driven monitoring of multimode continuous processes: a review. Chemometr. Intell. Lab. Syst. **189**(April), 56–71 (2019). https://doi.org/10.1016/j.chemolab.2019.03.012
74. J.F. Raskin, An introduction to hybrid automata, in *Handbook of Networked and Embedded Control Systems* (2005), pp. 491–517. https://doi.org/10.1007/0-8176-4404-0_21
75. I. Saberi, F. Faghih, F.S. Bavil, A passive online technique for learning hybrid automata from input/output traces (2021). arXiv preprint arXiv:2101.07053
76. M.A. Saez, F.P. Maturana, K. Barton, D.M. Tilbury, Context-sensitive modeling and analysis of cyber-physical manufacturing systems for anomaly detection and diagnosis. IEEE Trans. Autom. Sci. Eng. **17**(1), 29–40 (2020). https://doi.org/10.1109/TASE.2019.2918562. https://ieeexplore.ieee.org/document/8894669/
77. A.P. Sage, J.L. Melsa, *System Identification*, vol. 80 (Elsevier Science & Technology, 1971)
78. J. Saives, G. Faraut, J.J. Lesage, Automated partitioning of concurrent discrete-event systems for distributed behavioral identification. IEEE Tran. Autom. Sci. Eng. **15**(2), 832–841 (2018). https://doi.org/10.1109/TASE.2017.2718244
79. C. Sammut, S. Hurst, D. Kedzier, D. Michie, Learning to fly, in *Proceedings of the Ninth International Workshop on Machine Learning* (Aberdeen, Scotland, United Kingdom, 1992), pp. 385–393
80. C. Schlick, Simulation of rule-based behavior for a multimodal interaction task with stochastic petri nets. Proc. Hum. Fact. Ergon. Soc. Ann. Meeting **44**(6), 604–607 (2000). https://doi.org/10.1177/154193120004400616. http://journals.sagepub.com/doi/10.1177/154193120004400616
81. A. Singhal, D.E. Seborg, Clustering multivariate time-series data. J. Chemometr. **19**(8), 427–438 (8 2005). https://doi.org/10.1002/cem.945. http://doi.wiley.com/10.1002/cem.945

82. R. Sommer, V. Paxson, Outside the closed world: on using machine learning for network intrusion detection, in *2010 IEEE Symposium on Security and Privacy* (IEEE, 2010), pp. 305–316. https://doi.org/10.1109/SP.2010.25. http://ieeexplore.ieee.org/document/5504793/
83. R. Srinivasan, C. Wang, W.K. Ho, K.W. Lim, Dynamic principal component analysis based methodology for clustering process states in agile chemical plants. Ind. Eng. Chem. Res. **43**(9), 2123–2139 (4 2004). https://doi.org/10.1021/ie034051r. https://pubs.acs.org/doi/10.1021/ie034051r
84. T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, G. Tr, Wearable activity tracking in car manufacturing. IEEE Pervasive Comput. **7**(2), 42–50 (2008). https://doi.org/10.1109/MPRV.2008.40. http://ieeexplore.ieee.org/document/4487087/
85. A. Summerville, J. Osborn, M. Mateas, CHARDA: causal hybrid automata recovery via dynamic analysis, in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (International Joint Conferences on Artificial Intelligence Organization, California, 2017), pp. 2800–2806. https://doi.org/10.24963/ijcai.2017/390. https://www.ijcai.org/proceedings/2017/390
86. T. Tapia-Flores, E. Lopez-Mellado, A.P. Estrada-Vargas, J.J. Lesage, Discovering petri net models of discrete-event processes by computing T-invariants. IEEE Trans. Autom. Sci. Eng. **15**(3), 992–1003 (2018). https://doi.org/10.1109/TASE.2017.2682060
87. M.A. Umer, A. Mathur, K.N. Junejo, A. Adepu, Generating invariants using design and data-centric approaches for distributed attack detection. Int. J. Critic. Infrastructur. Protect. **28**, 100341 (2020). https://doi.org/10.1016/j.ijcip.2020.100341. https://linkinghub.elsevier.com/retrieve/pii/S1874548220300056
88. D. Underwood, *Types of Industrial Automation Systems* (2018). https://kingstar.com/types-industrial-automation-systems/
89. S. Verwer, *Efficient Identification of Timed Automata: theory and Practice*, Ph.D. thesis (Delft University of Technology (TU Delft), 2017)
90. J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, A. Serebrenik, Process discovery using integer linear programming, in *Applications and Theory of Petri Nets* (Springer, Berlin, Heidelberg, 2008), pp. 368–387. https://doi.org/10.1007/978-3-540-68746-7_24. http://link.springer.com/10.1007/978-3-540-68746-7_24
91. W. van der Aalst, *Process Mining*, 2nd edn (Springer, Berlin, Heidelberg, 2016). https://doi.org/10.1007/978-3-662-49851-4, http://link.springer.com/10.1007/978-3-662-49851-4
92. X. Wang, X. Wang, Z. Wang, F. Qian, A novel method for detecting processes with multi-state modes. Control Eng. Pract. **21**(12), 1788–1794 (2013). https://doi.org/10.1016/j.conengprac.2013.08.016
93. W. Wolf, Cyber-physical systems. Computer **42**(3), 88–89 (3 2009). https://doi.org/10.1109/MC.2009.81. http://ieeexplore.ieee.org/document/4803901/
94. J. Yu, S.J. Qin,: Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. AIChE J. **54**(7), 1811–1829 (2008). https://doi.org/10.1002/aic.11515
95. X. Zheng, C. Julien, M. Kim, S. Khurshid, Perceptions on the state of the art in verification and validation in cyber-physical systems. IEEE Syst. J. **11**(4), 2614–2627 (2017). https://doi.org/10.1109/JSYST.2015.2496293
96. F. Zhu, P.J. Antsaklis, Optimal control of hybrid switched systems: a brief survey. Discr. Event Dyn. Syst.: Theory Appl. **25**(3), 345–364 (2015). https://doi.org/10.1007/s10626-014-0187-5

# Vulnerability Management in IIoT-Based Systems: What, Why and How

**Geeta Yadav, Kolin Paul, and Praveen Gauravaram**

**Abstract** Industrial Control Systems (ICS) are characterized by large numbers of tightly integrated, interdependent, and heterogeneous components in a network. They act as a base system for safety and mission-critical Industrial Internet of Things (IIoT) applications such as smart grids, nuclear power plants, process control systems and robotics systems. The complex ICS, e.g., Supervisory Control and Data Acquisition (SCADA), consists of many interdependent subsystems. Modern SCADA systems are an amalgam of IIoT and legacy systems. IIoT is essentially a realization of advances in the connectivity of hardware and data networks that SCADA provides. Therefore, modern SCADA has evolved as a use case of IIoT, wherein IIoT improves industrial productivity by analyzing data generated by SCADA systems. The modernization of the SCADA system, standardization of communication protocols and almost ubiquitous interconnectivity courtesy for IIoT has drastically increased the attack surface of the SCADA system. Systematic Vulnerability Management (VM) of these attack surfaces minimizes risks and impacts associated with vulnerability exploitation. In this chapter, we first find the correlation between the IIoT and SCADA systems, followed by security challenges faced by IIoT-based systems. Then we highlight the role of VM in securing the critical systems, followed by the study of the state-of-art approaches for VM. After that, we discuss some future research directions for developing techniques for efficient VM. The chapter underscores the design challenges and research opportunities for efficiently managing the increasing vulnerabilities.

G. Yadav (✉) · K. Paul
Indian Institute of Technology Delhi, New Delhi, India
e-mail: geeta@cse.iitd.ac.in

P. Gauravaram
Tata Consultancy Services (TCS), Brisbane, Australia

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_3

## 1 Introduction

Over the years, an increase in the number of cyberattacks targeting the Industrial Control Systems (ICS) such as Supervisory Control and Data Acquisition (SCADA) systems have drawn the security researchers' attention towards these system's securities. There are several real-world cyber attacks on ICS infrastructures as discussed below.

**Ransomware on US fuel pipeline** In 2021, a ransomware attack encrypted critical data of Information Servers used in the SCADA stack of US Colonial Pipeline company [45]. As a consequence, Colonial pipeline company suspended all of the pipeline's operations as a precaution and to prevent further cascading impact. The adversaries stole nearly 100 gigabytes of data which led the company to pay 75 Bitcoin ($ 5 Million) to get the decryption tool due to a single compromised password. The pipeline shutdown impacted fuel shortages at airports and filling stations, resulted in canceling flights and panic fuel buying.

**Polish airline attack** [29] was due to Distributed Denial of Service (DDoS) attack, which overwhelms a network with traffic. The security expert took five hours to resolve the issue, leading to 10 flights cancellation and delays of around 15 flights at Warsaw Chopin airport.

The digital cyber-weapon **Stuxnet** [19] targeted at SCADA systems in 2010 is considered to be the most sophisticated cyber-attack. A malware jumped across air-gapped networks and damaged nuclear centrifuges of Iranian enrichment plants exploiting four unpatched zero-day Microsoft vulnerabilities used for self-replication and privilege escalation. Stuxnet damaged the centrifuges used in the uranium enrichment process by modifying their rotor speed. Vibrations and distortions caused by significant and sudden changes in their speed destroyed a thousand centrifuges, leading to less enriched uranium production.

**Ukraine power grid attack** [19, 46] in December 2015, where hackers hacked the information systems of three energy distribution companies using BlackEnergy malware. It resulted in rolling power outages for 1–6 h and affected 225,000 users.

In **German Steel Plant cyberattack** [28], the attackers gained unauthorized access to the mill's control systems using spear-phishing social engineering attacks. It led to an abnormal and unscheduled shutdown of the furnace, resulting in massive physical damage to the steel plant.

These incidents demonstrate the impact of a cyberattack by a determined adversary on such Critical Infrastructure (CI). Such cyberattacks could affect the availability of the software running on the device or can be used to reveal the running application's secrets. Devices under attack could stop working, behave differently, or be

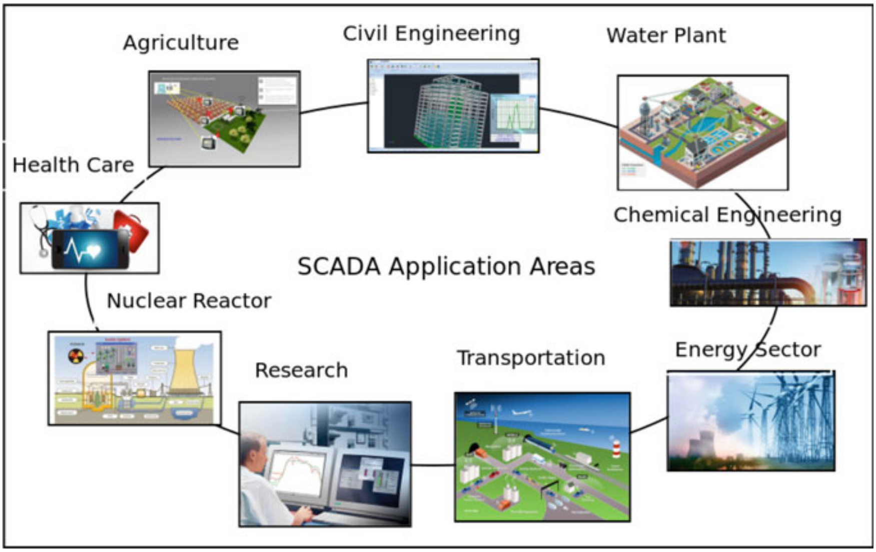


**Fig. 1** SCADA application areas [60]

leveraged to pose DDoS attacks either exploiting zero-day or reported yet unpatched vulnerabilities in a system. Moreover, these attacks have been led due to vulnerable SCADA systems by exploiting multiple vulnerabilities on different systems, generally referred to as Multi-host Multi-stage (MhMs) cyberattacks. SCADA systems, a type of ICS, are characterized by large numbers of tightly integrated, interdependent and heterogeneous components in a network [32]. The smooth and genuine operation of the SCADA framework is one of the key concerns for enterprises because the outcome of the breakdown of the SCADA system may range from financial loss to environmental damage to loss of human life [12]. These systems act as the base for safety and mission-critical infrastructures such as smart grids, nuclear power plants, process control systems and robotics systems [60]. These systems have become an essential part of automated control and monitoring of CI such as agriculture, healthcare, nuclear reactor, transportation, energy sector, civil and chemical engineering, water plants, research etc., as depicted in Fig. 1. Considering the significance of SCADA and ICS security that underpin critical national infrastructure, US Government offered policy recommendations for synchronizing foreign and domestic cyber security efforts and realizing a resilient and secure infrastructure [54].

*Evolution of SCADA systems*: Modern SCADA systems have evolved from standalone systems into sophisticated, complex and open systems connected to the Internet. With Industry 4.0/Industrial Internet of Things (IIoT) evolution, modern SCADA systems have adopted Cyber-Physical System (CPS)/IIoT, cloud technology, big data analytics, artificial intelligence and Machine Learning (ML). IIoT, generally defined as a sub-set of the Internet of Things (IoT) in terms of usage, covers the domains

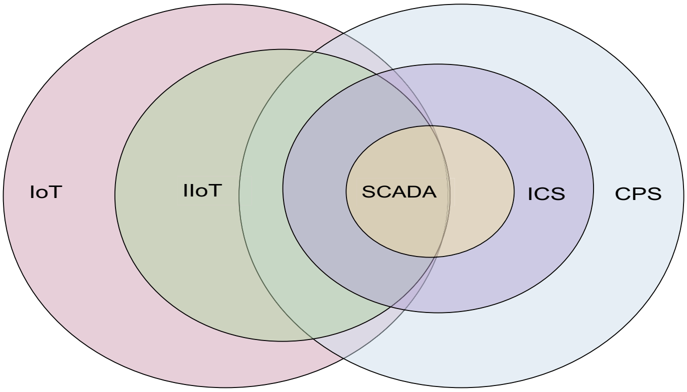


**Fig. 2** IIoT and SCADA

of machine-to-machine and industrial communication technologies with automation applications. IIoT paves the way for a better understanding of the manufacturing process, enabling efficient and sustainable production. IIoT allows a higher degree of automation by using cloud computing and data analytics to refine and optimize the process controls [9]. It further enables efficient interaction between the physical world and the cyber world, usually addressed as a CPS. ICS is the critical component to realize CPS. ICS provides control and monitoring functionality in manufacturing and industries.

**Correlation of IIoT and SCADA systems**: In Fig. 2, we demonstrate the overlap of IoT, IIoT, SCADA, ICS and CPS systems. IIoT is a subset of IoT. ICS such as SCADA is used to control CPS. Modern SCADA has been evolved into a connected IIoT-based system i.e., modern SCADA systems are an amalgam of IIoT and legacy systems as shown in Fig. 3. IIoT is essentially a realization of advances in the connectivity of hardware and data networks that SCADA provides. From the security perspective, the differences between them is not important. Therefore, in this chapter, we consider SCADA systems as a use-case for IIoT-based systems. We use IIoT-based SCADA systems and IIoT-based systems interchangeably.

In brief, integrating these technologies has significantly improved interoperability, eased maintenance and decreased the infrastructure cost. Therefore, modern SCADA systems are leading to a near real-time environment. Although IIoT improves the reachability in ICS, enhances data analytics, assuring ease of access and decision making, it also opens the ICS environment to attackers [14, 60]. The design of IIoT-based SCADA introduces multiple entry points to an isolated system, which is used to protect itself via air-gapping and risk avoidance strategies.

The Confidentiality, Integrity, and Availability (CIA) triad security model provides an excellent way to demonstrate the best practices to protect the data on the network. For the SCADA system, the security goal is generally the data availability that is the reverse of the prioritized security goals for traditional Information Tech-

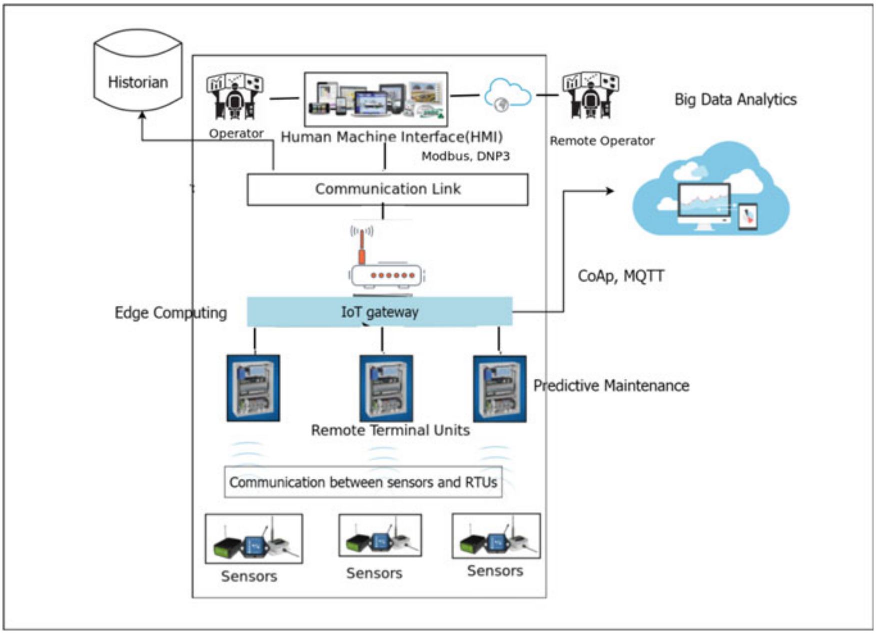


**Fig. 3** IIoT-based SCADA [60]

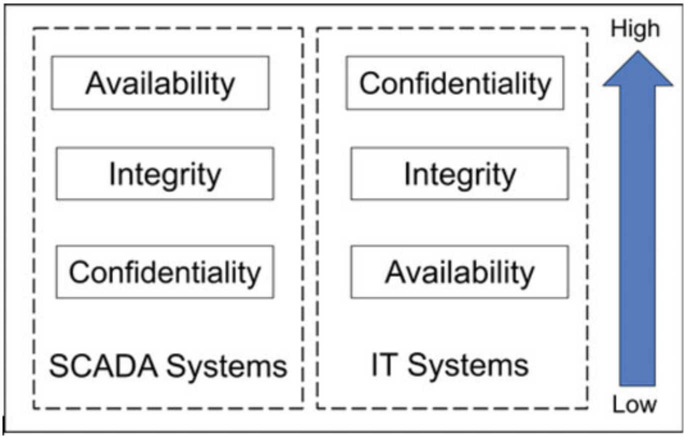


**Fig. 4** Priority order for SCADA and general IT

nology (IT) systems, as shown in Fig. 4. Therefore, downtime-constraints security is considered while implementing IIoT-security solutions. The ICSs are also called Operational Technology (OT) devices that control the physical world, while IT systems manage data [6]. Therefore, attackers generally target interrupting the SCADA system availability, causing production loss, financial loss, data loss, system damage, etc., hence tremendously affecting the economy, safety and security of a nation. An attacker needs to think outside the normal operating procedures to discover the unusual behavior, thus identifying vulnerabilities resulting in unauthorized access. The attacker needs only a single security hole, while a defender must defend against all possible security holes. Therefore, the defender needs to be more competent to compete with an attacker. Developing rigorous security layers can help to minimize the impact of attacks. A large number of vulnerabilities in various domains are reported to National Vulnerability Database (NVD) [38] each year. In NVD, 18,103 new vulnerabilities were reported in 2020 itself. With the integration of IIoT and legacy SCADA, the vulnerabilities reported to other domains are also applicable to IIoT-based SCADA [53], in a characterization study of ICS patching behavior, observed a patch delay of approximately 60 days after vulnerability disclosure for 50% of ICS devices. This lack of intime patching gives adversaries ample time to exploit these systems' publicly disclosed vulnerabilities.

Hence, the management of ICS security is becoming a major prevalent challenge due to an increase in system complexity and interdependencies. The progressive nature of ICS further complicates the scenario. On the one hand, the increasing complexity of software usually translates into more software flaws and vulnerabilities to fix. On the other hand, system threats continuously evolve, changing the risk outlook as new vulnerabilities and attack vectors emerge. In brief, to minimize the potential impact of successful cyberattacks, Vulnerability Management (VM) plays a pivotal point in any strategy for system security management.

In this chapter, we highlight the role of VM in securing critical systems, followed by the study of the state-of-art approaches for VM in Sect. 2. After that, we figure out the future research direction for developing techniques for efficient VM in Sect. 4. The chapter concludes by underscoring the design challenges and research opportunities for efficiently managing the increasing vulnerabilities in Sect. 5.

## 2 Vulnerability Management

VM is an indispensable part of managing an organization's safety and security. VM allows an organization to get a continuous overview of vulnerabilities in their OT environment. It is generally characterized as a cyclical process of five stages, i.e., Vulnerability discovery, Vulnerability analysis, Vulnerability prioritization, Vulnerability remediation and Vulnerability verification and monitoring.

**What is VM?** Strategic vulnerability management reduces the risk associated with vulnerability exploitation. In a generic term, VM tries to answer the following questions:

1. Do vulnerabilities exist on organizations' assets? If yes, what are they?
2. What are the characteristics of the discovered vulnerabilities?

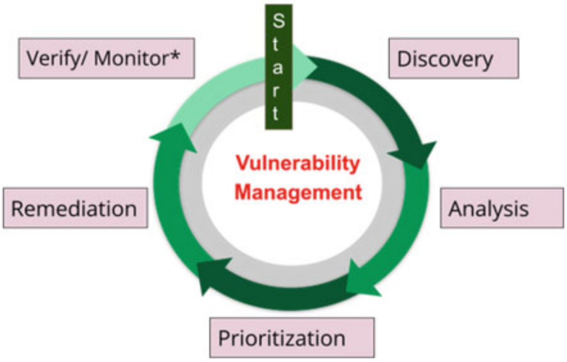


**Fig. 5** Generic vulnerability management lifecycle (*Extended stage)

3. What are the efficient strategies to fix the vulnerabilities so that the vulnerability exploitation's impact is minimal? Is there a critical need to patch all vulnerabilities?
4. What are the mechanisms for efficient and safe patch deployment?
5. Are the systems working normally post-patch deployment? Also, what vulnerabilities can not be patched yet have high risk? What are the monitoring strategies for unpatched vulnerabilities?

**How is VM performed?** VM is a cyclical practice of discovering, analyzing, prioritizing, remediating and verifying/monitoring possible exploitation of vulnerabilities in operating systems (OSs), enterprise applications, browsers and end-user applications, as shown in Fig. 5. In the first step, the vulnerabilities are generally discovered using a vulnerability scanner such as Nessus [50] and Nozomi networks [39]. Then in the second stage, the vulnerability scanner generates a consolidated report of possible known vulnerabilities. The security experts analyzed the report to prioritize the vulnerabilities based on their expertise and network knowledge in the third stage. The high severity vulnerabilities are selected for patching and the respective patch is deployed in the fourth stage. Once the vulnerabilities have been identified and resolved, consistent follow-up audits are required to ensure the mitigation is working in the fifth stage. This stage of vulnerability management is called the verification stage that helps to maintain transparency and accountability over the remediation process. Further, there can be two scenarios (i) the patch[1] is not available, (ii) a patch can not be applied to the system due to resource constraints or availability requirements. This gives an adversary ample time to exploit those vulnerabilities. Therefore, it is highly recommended to monitor the system to detect ongoing exploitation on time to minimize the potential damage. We extend the standard VM cycle by monitoring such a set of vulnerabilities in the fifth stage.

**Why is VM needed?** In brief, the lack of an appropriate plan for cyber-securing the assets in IIoT-based SCADA can cause organizations to have high risks of losing

[1] A security patch is applied to the system to fix the vulnerability to prevent successful exploitations.

revenue and reputation. VM is crucial to prioritize possible threats, reduce their attack surface and minimize the potential impact of cyber-attacks.

### *2.1 Challenges of IIoT-Based Systems for VM*

The challenges of securing the IIoT-based systems are as follows:

1. One of the critical things that enterprises need to consider ahead of VM in IIoT is constant, uninterrupted availability of the systems except for scheduled maintenance downtime [38]. The security solutions should either work concurrently without interfering with the system's functionality, or any change to the system should only be deployed at the scheduled downtime. This raises constraints on efficiently managing the VM cycle. Among the five stages of VM, patch deployment is the crucial phase, which hinders the system's functionality. Therefore, it becomes challenging for system administrators to effectively manage the scheduled downtime to fix the vulnerabilities issues.
2. The second challenge arises due to the blend of legacy and IIoT infrastructures [25], leading to increased attack surface and increased number of attack paths to exploit the legacy vulnerabilities. This leads to legacy vulnerabilities being targeted by the attackers [48].
3. The third challenge for system administrators is to monitor and control the end-to-end security of such large and complex critical industries [37].
4. The proposed solutions for efficient VM should consider the downtime constraints to take care of the various challenges mentioned above.

This short discussion presented above helps to identify the gap in the state-of-the-art leading the research contributions mentioned in the next section.

## 3 Tools and Techniques for Systematic VM

In this section, we discuss Tools and techniques for each stage of systematic VM in detail.

### *3.1 Vulnerability Discovery*

Discovering security vulnerabilities in software is a demanding task that requires significant human efforts. Vulnerability discovery is often the liability of software testers before release and white-hat hackers using bug bounty programs after the software is released. However, testers typically aim to find bugs related to performance and functionality with little focus on the security bugs due to the lack of expertise needed to discover security bugs. In [30] observed that only 40% of the tester have formal training in software engineering practices. Apart from that, black-hat hackers also identify vulnerabilities and later exploit them to gain economic or political benefits. The bug-bounty programs offer bounties in terms of money or recognition to vulnerability discoverers [18]. Therefore, vulnerability discovery is a competition between software testers and white-hat hackers vs. black-hat testers. Discovering vulnerabilities before the software release not only save time, money, a company reputation but also provides users protection and concerns regarding the patch deployment, especially in CPS, where the availability of the systems is the primary concern. Software development with the consideration of security reduces the reported vulnerabilities. Over time, vulnerability discovery tools have evolved to discover vulnerabilities automatically. However, human intelligence acts as a supplement to these tools.

A vulnerability discovery process can be divided into five stages: information gathering, program understanding, attack surface exploration and vulnerability recognition, and reporting [52]. In the information gathering state, the major goal is to understand prior efforts and the base technologies for the program. It plays a critical role in deciding whether to expend additional effort or resources or move on to a different target. In the program understanding state, the hackers attempt to learn the program behavior and its interaction with users and the network. After discovering the program's functionality, the hacker tries to identify the attack surface. This step leads to identifying resources that can be manipulated to influence the program execution and identification of critical components of the program. In the vulnerability recognition step, system administrators explore malicious activities and pass malicious input using automated tools to identify the malicious states of software. An iterative process of program understanding, attack surface exploration and vulnerability recognition leads to identifying vulnerabilities in the system. A comprehensive report is generated in the last stage, including the vulnerability reproduction steps, which the developer later uses to generate the patches. The skilled testers perform penetration testing to identify the vulnerabilities in the system. Penetration testing (commonly known as pentesting) is an authorized simulated cyberattack on a computer system to check for exploitable vulnerabilities. The penetration tests can be performed against the system from inside or outside to study all possible attackers' strategies. Each penetration test specifies guidelines and recommendations to address the identified issues. It is generally categorized into three types: black-box, grey-box, and white-box [26]. In the case of black-box testing, no information is available to the attacker. However, in the case of grey-box testing, basic information about the network is available to the attacker. In white-box testing, detailed system information, network architecture is available to the tester [7]. Since attackers access the target system from an outside network, the black-box testing results are the most realistic pentesting technique. Most widely used tools for pentesting, such as Nmap Metasploit, Burp suite Sqlmap, subfinder are freely available on Kali Linux. A thorough penetration testing when implementing IIoT architecture will reduce the reported vulnerabilities after the software is released. In large-scale IIoT networks, manually testing each system is challenging under resource-constrained scenarios, hence the researchers focus on automated security analysis solutions.

A manual penetration testing approach was proposed by Denis et al. [13] performing individual system penetration testing using the tools within the Kali Linux on smartphones and computers. The attacks performed were traffic sniffing, Man-in-the-Middle attack, hacking phone Bluetooth, remote desktop and open ports, etc. The primary focus of the work is to demonstrate penetration testing in a simplistic way. On the same line of work [51], developed PENTOS, a pentest tool specially designed for IoT devices to increase security awareness. PENTOS is a Graphical User Interface (GUI)-based tool on Kali-Linux, which first gathers the target system wirelessly followed by performing attacks such as web attacks and password attacks to get unauthorized access, followed by a report generation for successful attacks. PENTOS also has security guidelines for Open Web Application Security Project's top 10 vulnerabilities [41] to increase awareness [13, 51] provide practical experience of penetration testing. However, they do not demonstrate how to apply them on heterogeneous IoT nodes. Moreover, both the works are limited to a fixed set of attacks and are not scalable to a large IIoT network. With the increase in the complexity and size of the IIoT network, pentesting each and every system is a very challenging task. Therefore, researchers have focussed on using penetration graphs first to analyze the feasibility of exploitation. It facilitates the testers' analysis of the target network and provides a reference for executing penetration testing. In this direction, [56] proposed an automatic penetration graph generation algorithm combining the penetration graph generation method with the CVSS information. The authors made heuristics for generating the penetration graph that if a vulnerability has a CVSS score in the range [7–10], it will lead to admin privilege. However, they did not evaluate their framework in terms of scalability and IIoT applicability. AlGhazo et al. [1] proposed a framework that enlists a set of all possible sequences in which atomic-level vulnerabilities can be exploited to compromise specific system-level security given the networked system description. The traditional penetration testing systems are targeted to the pentesting of a system individually, which fails to detect MhMs attacks. This highlights an urgent need for new algorithms, tools, and frameworks to secure such resource-constrained devices. Koroniotis et al. [27] proposed a DL-based penetration testing framework using LSTM enabled vulnerability identification to detect the scanning attacks. The authors used Nessus, Zeek and Scapy to collect the training data by performing fuzzing scanning attacks against the network-enabled components of the smart airport-based testbed. This led to the generation of network traffic that was gathered, processed and labeled.

**Future directions**: In Table 1, we compare state-of-the-art vulnerability discovery approaches. We observed that most vulnerability discovery approaches focus on isolated system testing with a little focus on user-friendly GUI. These approaches will not detect the possible attacks exploiting MhMs vulnerabilities. Moreover, the penetration report only mentions the vulnerabilities reported, without further analysis, which are the critical vulnerabilities, which systems are critical in the network and

**Table 1** Vulnerability discovery: summary of the related work

| Research work | Tools used | Vulnerability databases used | Attacks performed | MhMs attack-paths | Critical path, node, vulnerability selection | GUI-based |
|---|---|---|---|---|---|---|
| Denis et al. [13] | Tools within the Kali Linux suite particularly Metasploit, Wireshark, Ettercap | ✗ | Traffic sniffing, man-in-the-middle attack, hacking phone bluetooth and remote desktop and open ports | ✗ | ✗ | ✗ |
| Visoottiviseth et al. [51] | Tools within the Kali Linux suite | ✗ | Password attack, web attack and wireless attack | ✗ | ✗ | ✓ |
| Xueqiu et al. [56] | Attack graph | Severity score provided by CVSS | ✗ | ✓ | ✗ | ✗ |
| Al Ghazo et al. [1] | Attack graph | ✗ | Remote code execution, unquoted servicepaths, user credentials construction, cross-site scripting, authentication token/cookie | ✗ | ✗ | ✗ |
| Koroniotis et al. [27] | Deep learning & Nessus, Zeek and Scapy to perform fuzzing scanning attacks to gather data for training | ✗ | Scanning attacks | ✗ | ✗ | ✗ |

the most likely exploited attack paths. This analysis helps the system administrators to take proactive measures to secure the network.

### *3.2 Vulnerability Analysis*

After identifying the vulnerabilities in the network using network scanners, penetration testing, etc., the next step of VM is to assess the vulnerabilities. A systematic and strategic assessment of a vulnerability would provide an actual severity and impact leading to an efficient resource allocation strategy. The NVD uses CVSS to analyze and assign a severity score to a vulnerability in the range [0, 10]. The vulnerabilities are analyzed based on their basic characteristics (such as Attack complexity, Attack vector, Privilege needed), temporal characteristics (such as Exploit Code Maturity, Remediation Level, Report Confidence) and environmental characteristics. Weighted Impact Vulnerability Scoring System (WIVSS) [49] is proposed to achieve higher diversity and accuracy of severity scores. WIVSS uses factors similar to CVSS, i.e., attack vector, attack complexity, authentication, confidentiality impact, integrity impact and availability impact. However, it uses different weights for the impact metrics (confidentiality impact, integrity impact and availability Impact) compared to the CVSS.

Phillips et al. [43] proposed a graph-based vulnerability analysis system, where a node represents a stage of attack and edge represents the transitions between the attack stages for network-vulnerability analysis considering internal and external attackers. The analysis system needs a common attack database with respective network configuration and topology configuration is analyzed. The level of effort is calculated by combining the probability of success on the edges. The likelihood of success is proportional to attack-path length. The major limitation of the work lies in the need for atomic steps of attacks. In a practical case, an attacker does not always follow a fixed set of patten. Moreover, the authors only presented a brief idea about the analysis system with no implementation and scalability analysis.

Ammann et al. [5] proposed a scalable vulnerability analysis approach by considering an assumption of monotonicity, i.e., the precondition of an exploit remains the same irrespective the attacker has exploited another vulnerability. The goal is achieved by combining the attacker access privilege, network connectivity and vulnerability in a common attribute, reducing the attack graphs' complexity.

**Future directions**: CVSS and WIVSS do not consider the domain characteristics while scoring the vulnerabilities. Therefore, directly using CVSS severity score and analysis may not give the exact severity of a vulnerability. Hence, extending the CVSS vulnerability analysis is necessary by considering the environment and network characteristics for deploying further security measures.

### *3.3 Vulnerability Prioritization*

With the expansion of networks due to IIoTization, more and more IIoT devices are connected to the Internet. Hence, there is a drastic increase in the number of vulnerabilities reported on these systems. Currently, NVD contains more than 1.60 lakhs vulnerabilities, out-of-which 18,767 vulnerabilities were reported in 2020 itself. Patching each vulnerability is a very challenging task. However [21], studied the ratio of vulnerability exploited and vulnerability reported for 2009–2018. 76 k vulnerabilities were reported to NVD in the mentioned period, out of which about 12.8% (9.7/76 k) of all vulnerabilities had their published exploit code. A key observation is that only about 5% (4.2/76 k) vulnerabilities were exploited. This shows that not all vulnerabilities are exploited, nor all vulnerabilities can be patched in a resource-constrained scenario. Hence, vulnerability prioritization should be considered.

To efficiently handle these scenarios in a resource-constrained environment, industries prioritize vulnerability patching using crude heuristics based on limited data. Hence, many known vulnerabilities are breached by attackers for which the patch was already available. It raises a few challenges to the system administrators:

1. Suppose we patch all the vulnerabilities of the network. In that case, resources are consumed on the low-severity vulnerability, which has less probability of exploitability and low impact, even if they got exploited.
2. In another scenario, if we patch a few critical-severity vulnerabilities, it may be an economical, efficient strategy but may lead to other high-risk vulnerabilities, including MhMs exploitation.

In brief, vulnerability prioritization is a practice to balance resource availability and exploitation impacts with a large amount of discovered vulnerabilities. The vulnerability prioritization should be strategic and efficient.

Game theory has been used widely in capturing the strategic interactions between the intelligent agents, i.e., the attacker and the defender, where the payoff of each depends not only on their own action but also on other players' actions. Apart from game theory, graph theory is also used to find an optimized strategy. The expert analysis also helps to understand the severity of a vulnerability. Next, we discuss related work in each category, i.e., expert analysis based, graph theory-based and game theory-based approaches in detail.

**Expert analysis based vulnerability prioritization approaches**: The CVSS is an indicator of true vulnerability severity. CVSS is used by nexpose [44] vulnerability management tool to rank the vulnerabilities. However, the severity score provided by CVSS is static and has not changed over time. These scores are standard for all systems and can be improved by considering temporal and environmental metrics with base metrics [17]. WIVSS [49] is proposed to achieve higher diversity and accuracy of severity scores. WIVSS uses factors similar to CVSS, i.e., attack vector, attack complexity, authentication, confidentiality impact, integrity impact and availability impact. However, it uses different weights for the impact metrics (confidentiality impact, integrity impact and availability impact) compared to the CVSS.

**Graph-based vulnerability prioritization approaches**: Graph-based vulnerability prioritizing approaches like SecureRank [35], Risk-Rank [3] and VULCON [16] provide a static ranking of patching order and they do not consider the behavior of an attacker. SecureRank defines a security metric based on the percentage of time a random attacker would spend endeavoring to exploit a vulnerability successfully. It takes network topology and vulnerability severity as inputs and returns defense probability for each subsystem. Defense probability denotes the probability of selecting a vulnerability on a particular subsystem for patching to reach the optimal state. Our framework in stage 3 establishes that it reaches a Nash equilibrium. The authors compared SecureRank with density, source and type-based prioritization and observed that SecureRank provides an effective and efficient patch prioritization approach. It prioritizes vulnerabilities based on a balance between immediate risk and the risk due to system interdependencies' cascading. The Risk-Rank algorithm captures the risk diffusion by using complex interaction over time. Risk-Rank is verified by using a case study based on the organization's conceptual structure, business units' risk dependencies and vulnerabilities. VULCON is a patch prioritization framework proposed for network security management. It is based on fundamental performance metrics, i.e., "time-to-vulnerability remediation" and "total vulnerability exposure". The proposed algorithm uses a mixed-integer multi-objective optimization algorithm to prioritize vulnerabilities for patching subject to the given resource constraints. However, the graph theory-based approaches fail to incorporate the attacker behavior, which plays a vital role in analyzing the possible impact of exploiting a vulnerability.

**Game theory-based vulnerability prioritization approaches**: Game theory-based approaches for patch prioritization [4, 10, 24, 47] incorporate attackers' behavior to better estimate the prioritization strategy.

Alshawish and Risk de Meer [47] proposed a game-theoretical model to optimize the security strategy of electricity distribution networks with vulnerable Distributed Energy Resource (DER) nodes. The authors consider an adversarial model for false data injection attacks to compromise vulnerable nodes. The impact of this attack in a smart grid on a defender includes the loss of voltage regulation and the cost of induced load control under supply-demand mismatch between the generator and distributor. The proposed greedy approach is formulated in a three-stage defender-attacker-defender game, (i) the defender first chooses a strategy to secure DER nodes (ii) the attacker will try to compromise the DER nodes (iii) the defender chooses the security investments strategy by controlling the loads and non-compromised nodes. The authors use a greedy approach to compute attacker-defender strategies and recommend optimal financial investments to secure the systems. Kamdem et al. [24] proposed a two-player zero-sum Markov game to identify the optimal strategy to disconnect vulnerable services to slow down the attack.

Alshawish and Risk de Meer [4] proposed an integrated risk-based methodology for prioritizing possible vulnerability remediation activities by leveraging Time-To-Compromise (TTC) security metric. This model employs the network topology, attackers' capability and published vulnerability and exploit information. TTC is calculated by taking into account the total number of disclosed vulnerabilities, the number of high severity vulnerabilities, the number of low severity vulnerabilities, the total number of existing exploits, the expected time taken for identifying the zero-day vulnerability, the expected time taken for calculating the exploit and adversarial skill set. The authors provide a game-theoretic approach considering the stochastic nature of risk assessments across an electric power organization. The authors acknowledged that TTC-based models could convey misleading results due to the aggregation of anticipated features of a vulnerability. Chen et al. [10] proposed a bi-level optimization model under a game-theoretic framework to incorporate the interactions of a system administrator and an adversary. The interactions among cyber-physical elements are considered to determine cascading failure under potential attacks. The approach leads to optimal resource allocation by the system defender to maintain system reliability. However, the game theory-based approaches proposed earlier for patch prioritization consider only the single attacker-defender scenario, which is not pragmatic in all cases.

Apart from the above approaches [2], proposed an ML-based exploit prediction model leveraging vulnerability information from different databases, i.e., NVD, ExploitDB, ZDI and Dark Web (DW). The attacker behavior is integrated by considering the blogs/ posts for respective vulnerabilities on DW. However, the learning-based detection approaches may be deceived due to intentionally discussing the random vulnerabilities on DW by adversaries.

**Future directions**: In Table 2, we compare state-of-the-art vulnerability prioritization approaches based on architectural feature, vulnerability feature, patch dependencies, attacker feature and approach category. We observed that most approaches do not consider resource constraints, functional dependencies, patch dependencies and multiple defender-attackers practical scenarios. Incorporating an attacker's behavior plays a vital role in proper resource allocation and failure to consider the patch dependencies will lead to patch breaks while deployed. In this direction [57, 58], proposed a prioritization framework leveraging a game-theoretical model. However, the approach can be extended by considering different attacker strategies and network characteristics.

### *3.4 Vulnerability Remediation*

After the system administrators have analyzed and prioritized the vulnerabilities, the next phase is to deploy the patches. It is the most challenging stage of patch management due to the complexities of software arising from network inter-connectivity and inter-dependencies. Hence, a patch deployment may affect other dependent applications potentially. Apart from software dependencies, the patch dependencies hierarchy needs to be considered, i.e., if patch A depends upon patch B then before deploying patch A, the administrators need to deploy patch B. Another concern is the limited time between the patch availability and the exploit release, leading to the high probability of successful exploitation. This raises concern for the deployment of

**Table 2** Vulnerability prioritization: summary of the related work

| Research work | Architectural feature | Vulnerability feature | Attacker feature | Functional dependency | Multiple-attacker-defender | Patch dependencies |
|---|---|---|---|---|---|---|
| Rapid7-community [44] | ✗ | Attack vector, attack complexity, scope, impact (confidentiality, integrity, availability) | ✗ | ✗ | ✗ | ✗ |
| Spanos et al. [49] | ✗ | Weighted characteristics (Attack vector, attack complexity, scope, impact (confidentiality, integrity, availability) | ✗ | ✗ | ✗ | ✗ |
| Miura-Ko and Bambos [35] | Network topology | % of time a random attacker would spend trying to exploit | Random attacker | ✗ | ✗ | ✗ |
| Alpcan and Bambos [3] | Bipartite graph | ✗ | ✗ | ✗ | ✗ | ✗ |
| Farris et al. [16] | Mission criticality score | Total vulnerability exposure, Time-to-vulnerability remediation | ✗ | ✗ | ✗ | ✗ |
| Shelar and Amin [47] | Network model of radial electric distribution systems | Impact of attack in terms of loss of voltage regulation and cost of induced load control under supply-demand mismatch | Three-stage defender-attacker-defender game | ✗ | ✗ | ✗ |
| Alshawish and Meer [4] | Network topology | Time-to-compromise feature | All possible attack-paths to the target node | ✗ | ✗ | ✗ |
| Yadav and Paul [59] | Inter dependencies | Cost of defend, cost of attack and Impact of attack (NVD Database, CVSS score) | Single attacker | ✗ | ✗ | ✗ |
| Kamdem et al. [24] | Nework topology | CVSS severity score | Single attacker | ✗ | ✗ | ✗ |
| Chen et al. [10] | Node-link model | Cascading failure under malicious attacks | Single attacker | ✗ | ✗ | ✗ |

the patches as soon as they are available. A security patch should be well tested before deployment. It may sometimes break the service rather than repairing faulty patches that introduce issues like backward compatibility, interoperability issue, patch break and introduction of a new vulnerability. The presence of faulty patches increases the cost of patch deployment and service downtime. Hence, many system administrators often delay installing patches and keep using outdated software, leaving known vulnerabilities readily exploitable. However, in ICS systems, the patching is scheduled with consideration of the requirement of the system availability, pre-deployment testing and post-deployment testing. In brief, the difficulty in dealing with patch dependencies and the significant amount of human effort required for configuring a test environment to simulate a production-identical environment hinders automated patch deployment. Therefore, before deploying a patch, a deep analysis of the impact of patch deployment should be done. A sophisticated live patching technique has been proposed to reduce the service downtime or maintenance window [33]. However, their applicability in practice is minimal. Commonly available virtualization capabilities allow system administrators to perform a majority of the patchwork outside of the maintenance window by capturing the disk activities and replaying them during the actual maintenance window.

**Future directions**: A patch management policy that tests and applies suitable patches to all affected areas in an efficient and timely manner is crucial. A trustworthy remediation solution helps developers, security and devOps teams by keeping them in sync so that the entire vulnerability management process runs smoothly. With the decrease in system downtime and the need to keep systems updated highlights the need for advanced techniques for live patching.

### *3.5 Vulnerability Verification and Monitoring*

Once the vulnerabilities have been patched, the next stage of VM is to verify or test the deployed patches on these systems. The patch deployments are verified by monitoring the systems for unexpected service interruptions. Manual patch deployment verification approaches are challenging, error-prone and time-consuming in complex networks. There is a lack of automated tools to overview the state of the system post-patch deployment.

Moreover, due to system availability requirements, few critical vulnerabilities can not be patched in ICS systems. In those cases, system administrators need to monitor these systems to timely detect ongoing exploitations. If an ongoing attack is detected in the network, which may lead a path to the critical asset, suitable actions to stop the attack or reduce the attack's impact should be taken. There has been active research for more than a decade for using system logs to detect the anomalous behavior of a system using either rule-based strategies or ML-based techniques on a single system. However, only a few approaches focus on correlating the attack scenario on different systems to find the indications of compromise, which leads to the detection of an attack before it reaches its target. We study the related work into three categories as discussed below:

**Techniques for anomaly detection on a system using system logs**: Rule-based anomaly detection approaches [40, 64] are limited to detect specific scenarios with high accuracy, requiring domain expertise. [64] represented the Syslog behaviors using a combination of hidden Markov models followed by learning the model using a discounting learning algorithm. Oprea et al. [40] proposed a graph-theoretic framework based on belief propagation to detect advanced persistence threats infection. The ML-based anomaly detection approaches can be categorized as supervised and unsupervised learning-based approaches. Supervised learning-based approaches derive a model from the labeled training data, which generally label data either normal or anomalous. Chen et al. [11] presented a decision tree-based approach to diagnose failures on Internet sites. First, they trained the decision trees on the request traces. The training request traces data also included the request failure scenarios visible and labeled by the user. When tested on real-failure data from eBay (an eCommerce website) request traces, the proposed approach successfully identified 13 out of 14 failure cases. Liang et al. [31] applied Support Vector Machine (SVM) to predict failures in IBM BlueGene/L event logs[2]. The supervised-learning-based approaches need a large amount of labeled data to train the model. In an unsupervised-learning algorithm clustering approach, LogCluster utilizes the base idea to check if a particular log sequence has occurred or not [31]. Apart from these, program invariants were used to detect abnormal events. Initially, program invariants are being identified to learn the linear relationships between system events during the program execution. A log sequence that does not follow the program invariants is labeled as anomalous. The above ML approaches made a close-world assumption that the log set is finite and data will be stable over the period. However, in practice, log data may encounter previously unseen log sequences, decreasing the accuracy of the anomalous log detection. In this direction, Deeplog [15] is an online anomalous log detection approach using the LSTM model. The approach consists of three key modules, i.e., key anomaly detection, parameter anomaly detection and workflow construction. Deeplog is trained on normal data only and can adapt new log patterns on false positive detection. Zhang et al. [65] proposed an anomaly detection approach by utilizing an attention-based Bi-LSTM model. Meng et al. [34] highlighted to use of the semantics of the log messages rather than the indexes, which is generally used for anomaly detection to reduce the false positive. LSTMs have proved to be a promising solution to sequence and time-series related problems.

[2] The event logs are the events from OSs, applications or devices and are stored in a single cluster by the operating system. Events logged by the operating system are also called system logs.

**Techniques for ongoing attack detection on single system logs**: In this direction [8], proposed a rule-based model to detect targeted port scans, detection of Cross-Site Scripting (XSS) and SQL Injection (SQLI) attacks using access logs of Apache HTTP Server. Moh et al. [36] leveraged the features of both rule-based and learning-based approaches to detect the SQLI attacks using web server logs. A collaborative approach by combining intrusion detection at different layers, i.e., network, kernel and application, can increase the accuracy of attack detection as compared to individual detectors, without much degradation in performance [55]. An attack-story reconstruction approach proposed by Pei et al. [42] correlates the log graph utilizing logs from different levels on a single host. However, these approaches are limited to attack detection on a single system with knowledge of how they can be used for correlated attacks.

**Approaches for temporal and spatial correlation of attacks using logs**: In this direction [11], proposed a process query system based on control and estimation methods to correlate the distributed network events. Attack graph has been used for correlating attacks on MhMs attacks. However, manual construction of the attack graphs is challenging and error-prone. Few automatic attack-graph generation have been proposed in literature e.g. [1, 20, 22, 62, 63]. The attack-graph generation approaches either use their model checker or use a knowledge database of vulnerabilities and exploits, e.g., NVD, ExploitDb etc., to generate the pre-requisites and post-conditions related to exploitation steps. The pre-requisite and post-conditions of a vulnerability denote the conditions needed to exploit a vulnerability and the capability gained by exploiting it. However, these approaches [1, 20, 22, 62, 63] limit themselves to generate the attack paths only, i.e., they will not detect any ongoing attacks. Therefore, there is a need for an effective methodology to detect ongoing MhMs attacks timely.

**Future directions**: In Table 3, we compare state-of-the-art vulnerability monitoring approaches based on 'Data used', 'Technique used', 'Attacks detected', 'Detect attacks on single system' and 'Detect MhMs attacks'. We observe that except [11], the approaches are targeted to detect vulnerabilities exploitation on a single system. [11] approach lack the practical implementation and feasibility analysis.

**Table 3** Vulnerability monitoring: summary of the related work

| Research work | Data used | Techniques used | Attacks detected | Detect attacks on single system | Detect MhMs attacks |
|---|---|---|---|---|---|
| Yen et al. [64] | Log data | ✗ | Variation from the normal behaiour | ✗ | ✗ |
| Oprea et al. [40] | Web proxy logs | Belief propagation inspired from graph theory | APT infection attack | ✗ | ✗ |
| Chen et al. [11] | Request traces | Decision trees | Causes of failures | ✗ | ✗ |
| Liang et al. [31] | System logs | SVM and nearest neighbor method | Variation from the normal behaiour | ✗ | ✗ |
| Zhang et al. [65] | System logs | An attention-based Bi-LSTM model | Variation from the normal behaiour | ✗ | ✗ |
| Meng et al. [34] | System logs | Extracting semantic information using Template2Vec | Sequential and quantitative anomaly detection | ✗ | ✗ |
| Du et al. [15] | System Logs | LSTM-based deep learning | Denial of service attack, port scan, socially engineered attack | ✓ | ✗ |
| Wu et al. [55] | Logs from IDS Snort, Libsafe and sysmon | Graph-based and a Bayesian network based aggregation method | Buffer overflow, flooding and script-based attacks | ✓ | ✗ |
| Pei et al. [42] | DNS logs, Auditd logs, Firefox logs, Syslog | Graph analytics | A phishing email, watering-hole attack, trojan software, an unofficial patch containing malicious payloads | ✓ | ✗ |
| Jiang and Cybenko [11] | System events | Control and estimation methods | | ✓ | ✓ |
| Ba et al. [8] | Access log files of Apache web servers | Rule-set based approach | Web scan detections, SQL injection and XSS attacks | ✓ | ✗ |
| Moh et al. [36] | Web server logs | Hybrid approach (combine rule and larning based approach) | SQL Injection | ✓ | ✗ |

## 4 Reseach Directions

The researchers should aim at building techniques for an efficient VM.

1. The researcher should analyze the reported vulnerabilities specific to IIoT-based SCADA systems to understand better the type of attacks, the vulnerable components, and the vulnerabilities' impact.
2. A focus on developing frameworks to analyze and find a series of vulnerabilities in different systems that are needed to exploit to reach the target system. The framework should recommend a consolidated report of the vulnerable state of the system and all possible target paths to the critical node. The framework should be scalable to the IIoT network.
3. Not all vulnerabilities are always exploited by the attackers, and not all vulnerabilities can be patched due to the resource constraints such as people, infrastructure, tools and time available to patch every vulnerability. Also, ICSs such as SCADA have strict system uptime and availability requirements. These constraints place significant importance on the patch prioritization of networks and devices, which needs to be strategic and efficient.

   There is a need to develop a patch prioritization framework that is applicable to ICSs. The prioritization order should consider the architectural characteristics to understand the domain knowledge of the target system, vulnerability characteristics to embed the vulnerability severity, patch dependencies to avoid the patch break on deployment and attacker behavior to reflect a practical scenario. The framework should recommend a strategy for patching, which is optimal and effective considering resource constraints.
4. Moreover, the researcher should focus on designing and developing frameworks that correlate the evidence of an incident spread temporarily and spatially in the network. The framework should detect the ongoing exploitation of MhMs vulnerabilities on a system. In this direction, GloM has been presented to monitor MhMs attack [61].

## 5 Conclusion

In this chapter, we first discussed the correlation between SCADA systems and IIoT-based systems, followed by the need of VM for securing these systems. We discuss what is VM? why we need VM? how to perform VM? Afterward, we discuss the issues with state-of-the-art vulnerability management approaches. We observed that vulnerability discovery approaches focus on isolated system testing with a little focus on user-friendly GUI. These approaches will not detect the possible attacks exploiting MhMs vulnerabilities. Moreover, the penetration report only mentions the vulnerabilities reported, without further analysis, which are the critical vulnerabilities, which systems are critical in the network and the most likely exploited attack paths. This analysis helps the system administrators to take proactive measures to secure the network. We observed that most vulnerability prioritization approaches do not consider resource constraints, functional dependencies, patch dependencies and multiple defender-attackers practical scenarios. Incorporating an attacker's behavior plays a vital role in proper resource allocation and failure to consider the patch dependencies will lead to patch breaks while deployed. We also observed that the vulnerability monitoring approaches are targeted to detect the exploitation of vulnerabilities on a single system only. Hence fail to detect the ongoing MhMs attacks timely.

## References

1. A.T. Al Ghazo, M. Ibrahim, H. Ren, R. Kumar, A2G2V: automated attack graph generator and visualizer. in *Mobile IoT SSP'18*, vol. 3 (ACM, Los Angeles, CA, USA, 2018), pp. 1–6. https://doi.org/10.1145/3215466.3215468
2. M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, P. Shakarian, Patch before exploited: an approach to identify targeted software vulnerabilities, in *AI in Cybersecurity*, ed. by F.S. Leslie (Springer International Publishing, Cham, 2019), pp. 81–113. https://doi.org/10.1007/978-3-319-98842-9_4
3. T. Alpcan, N. Bambos, Modeling dependencies in security risk management, in *2009 Fourth International Conference on Risks and Security of Internet and Systems (CRiSIS 2009)* (2009), pp. 113–116
4. A. Alshawish, H. Risk de Meer, Risk mitigation in electric power systems: where to start? Energy Inform. **2**(1), 34 (2019)
5. P. Ammann, D. Wijesekera, S. Kaushik, Scalable, graph-based network vulnerability analysis, in *Proceedings of the 9th ACM Conference on Computer and Communications Security. CCS '02* (Association for Computing Machinery, Washington, DC, USA, 2002), pp. 217–224. https://doi.org/10.1145/586110.586140
6. A. Andreu, Operational technology security—A data perspective. Netw. Secur. **1**, 8–13 (2020). https://doi.org/10.1016/S1353-4858(20)30008-8
7. R. Ankele, S. Marksteiner, K. Nahrgang, H. Vallant, Requirements and recommendations for IoT/IIoT models to automate security assurance through threat modelling, security analysis and penetration testing, in *Proceedings of the 14th International Conference on Availability, Reliability and Security. ARES '19* (Association for Computing Machinery, Canterbury, CA, United Kingdom, 2019). https://doi.org/10.1145/3339252.3341482
8. S.M. Ba, F.O. Catak, E. Gül, Detection of attack-targeted scans from the apache HTTP server access logs. Appl. Comput. Inf. **14**(1), 28–36. https://doi.org/10.1016/j.aci.2017.04.002
9. H. Boyes, B. Hallaq, J. Cunningham, T. Watson, The industrial internet of things (IIoT): an analysis framework. Comput. Ind. **101**, 1–12 (2018). https://doi.org/10.1016/j.compind.2018.04.015
10. K. Chen, W. Fushuan, C.-L. Tseng, M. Chen, Z. Yang, H. Zhao, H. Shang, A game theory-based approach for vulnerability analysis of a cyber-physical power system. Energies **12**(15), 3002 (2019). https://doi.org/10.3390/en12153002
11. M. Chen, A.X. Zheng, J. Lloyd, M.I. Jordan, E. Brewer, *Failure Diagnosis Using Decision Trees* (2004), pp. 36–43
12. Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, K. Stoddart, A review of cyber security risk assessment methods for SCADA systems. Comput. Secur. **56**, 1–27 (2016). https://doi.org/10.1016/j.cose.2015.09.009

13. M. Denis, C. Zena, T. Hayajneh, Penetration testing: concepts, attack methods, and defense strategies, in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (2016), pp. 1–6. https://doi.org/10.1109/LISAT.2016.7494156
14. L.L. Dhirani, E. Armstrong, T. Newe, Industrial IoT, cyber threats, and standards landscape: evaluation and roadmap. Sensors **21**(11) (2021). https://doi.org/10.3390/s21113901
15. M. Du, F. Li, G. Zheng, V. Srikumar, DeepLog: anomaly detection and diagnosis from system logs through deep learning, in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17* (Association for Computing Machinery, Dallas, Texas, USA, 2017), pp. 1285–1298. https://doi.org/10.1145/3133956.3134015
16. K.A. Farris, A. Shah, G. Cybenko, R. Ganesan, S. Jajodia, VULCON: a system for vulnerability prioritization, mitigation, and management. ACM Trans. Priv. Secur. **21**(4) (2018). https://doi.org/10.1145/3196884
17. C. Fruhwirth, T. Mannisto, Improving CVSS-based vulnerability prioritization and response with context information, in *2009 3rd International Symposium on Empirical Software Engineering and Measurement* (2009), pp. 535–544. https://doi.org/10.1109/ESEM.2009.5314230
18. R. Hamper, Software bug bounties and legal risks to security researchers. Ph.D. thesis (2019)
19. Idaho-National-Laboratory, History of industrial control system cyber incidents (2018). https://www.osti.gov/servlets/purl/1505628. Accessed 04 May 2020
20. K. Ingols, R. Lippmann, K. Piwowarski, Practical attack graph generation for network defense, in *Proceedings of the 22nd Annual Computer Security Applications Conference. ACSAC '06* (IEEE Computer Society, Washington, DC, USA, 2006), pp. 121–130. https://doi.org/10.1109/ACSAC.2006.39
21. J. Jacobs, S. Romanosky, I. Adjerid, W. Baker, Improving vulnerability remediation through better exploit prediction. J. Cybersecur. **6**(1), tyaa015 (2020). https://doi.org/10.1093/cybsec/tyaa015. https://academic.oup.com/cybersecurity/article-pdf/6/1/tyaa015/33746021/tyaa015.pdf
22. S. Jajodia, S. Noel, B. O'Berry, Topological analysis of network attack vulnerability, in *Managing Cyber Threats: Issues, Approaches, and Challenges. Ed. by Vipin Kumar, Jaideep Srivastava, and Aleksandar Lazarevic* (Springer US, Boston, MA, 2005), pp. 247–266. https://doi.org/10.1007/0-387-24230-9_9
23. G. Jiang, G. Cybenko, Temporal and spatial distributed event correlation for network security, in *Proceedings of the 2004 American Control Conference*, vol. 2 (2004), pp. 996–1001. https://doi.org/10.23919/ACC.2004.1386701
24. G. Kamdem, C. Kamhoua, Y. Lu, S. Shetty, L. Njilla, A Markov game theoritic approach for power grid security, in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (2004), pp. 139–144. https://doi.org/10.1109/ICDCSW.2017.63
25. K. Keshav, S.S. Vijay, D.M. Lourenço, A. Anil Kumar, P. Plapper, Retrofitting of legacy machines in the context of industrial internet of things (IIoT), in *3rd International Conference on Industry 4.0 and Smart Manufacturing on Procedia Computer Science*, vol. 200 (2022), pp. 62–70. https://doi.org/10.1016/j.procs.2022.01.205. https://www.sciencedirect.com/science/article/pii/S1877050922002149
26. M.E. Khan, F. Khan, A comparative study of white box, black box and grey box testing techniques. Int. J. Adv. Comput. Sci. Appl. **3**(6) (2012). https://doi.org/10.14569/IJACSA.2012.030603
27. N. Koroniotis, N. Moustafa, B. Turnbull, F. Schiliro, P. Gauravaram, H. Janicke, A Deep learning-based penetration testing framework for vulnerability identification in internet of things environments (2021). arXiv: 2109.09259 [cs.CR]
28. R.M. Lee, M.J. Assante, T. Conway, German steel mill cyber attack. Ind. Control Syst. 1–15 (2014)
29. M. Lehto, Cyber security in aviation, maritime and automotive. Comput. Big Data Transp. 19–32 (2010)
30. T.C. Lethbridge, J. Diaz-Herrera, R.J. Jr., LeBlanc, J.B. Thompson, Improving software practice through education: challenges and future trends, in *2007 Future of Software Engineering. FOSE '07* (IEEE Computer Society, USA, 2007), pp 12–28. https://doi.org/10.1109/FOSE.2007.13

31. Y. Liang, Y. Zhang, H. Xiong, R. Sahoo, Failure prediction in IBM blueGene/L event logs (2007); In Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, X. Chen, Log clustering based problem identification for online service systems, in *Proceedings of the 38th International Conference on Software Engineering Companion. ICSE '16* (Association for Computing Machinery, Austin, Texas, 2016), pp. 102–111. https://doi.org/10.1145/2889160.2889232
32. Y. Lu, P. Witherell, A. Jones, Standard connections for IIoT empowered smart manufacturing. Manuf. Lett. **26**, 17–20 (2020). https://doi.org/10.1016/j.mfglet.2020.08.006
33. M. Maurer, David Brumley, Tachyon: tandem execution for efficient live patch testing, in *21st USENIX Security Symposium (USENIX Security 12)*. (Bellevue, WA, USENIX Association, 2012), pp. 617–630
34. W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, R. Zhou, LogAnomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs, in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization* (2019), pp. 4739–4745. https://doi.org/10.24963/ijcai.2019/658
35. R.A. Miura-Ko, N. Bambos, SecureRank: a risk-based vulnerability management scheme for computing infrastructures, in *2007 IEEE International Conference on Communications* (2007), pp. 1455–1460. https://doi.org/10.1109/ICC.2007.244
36. M. Moh, S. Pininti, S. Doddapaneni, T.-S. Moh, Detecting web attacks using multi-stage log analysis, in *2016 IEEE 6th International Conference on Advanced Computing (IACC)* (2016), pp. 733–738. https://doi.org/10.1109/IACC.2016.141
37. A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, A. Urbieta, End to end secure data exchange in value chains with dynamic policy updates, in *CoRR* (2022). arXiv: 2201.06335
38. C. Niesler, S. Surminski, L. Davi, Hera: hotpatching of embedded real-time applications, in *28th Network and Distributed System Security Symposium (NDSS)* (2021); NIST, National vulnerability database (2021). https://nvd.nist.gov/
39. Nozomi-Networks, Nozomi-networks (2021)
40. A. Oprea, Z. Li, T.-F. Yen, S.H. Chin, S. Alrwais, Detection of early-stage enterprise infection by mining large-scale log data, in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (2015), pp. 45–56. https://doi.org/10.1109/DSN.2015.14
41. OWASP-community, OWASP top ten (2021). https://owasp.org/www-projecttop-ten/
42. K. Pei, Z. Gu, B. Saltaformaggio, S. Ma, F. Wang, Z. Zhang, L. Si, X. Zhang, D. Xu, HERCULE: attack story reconstruction via community discovery on correlated log graph, in *Proceedings of the 32nd Annual Conference on Computer Security Applications. ACSAC '16* (Association for Computing Machinery, Los Angeles, California, USA, 2016), pp. 583–595. https://doi.org/10.1145/2991079.2991122
43. C. Phillips, L.P. Swiler, A graph-based system for network-vulnerability analysis, in *Proceedings of the 1998 Workshop on New Security Paradigms. NSPW '98* (Association for Computing Machinery, Charlottesville, Virginia, USA, 1998), pp. 71–79. https://doi.org/10.1145/310889.310919
44. Rapid7-community, Working with vulnerabilities (2021). https://docs.rapid7.com/nexpose/working-with-vulnerabilities/. Accessed 13 June 2021
45. J.R. Reeder, C.T. Hall, Cybersecurity's pearl harbor moment: lessons learned from the colonial pipeline ransomware attack (2021)
46. SANS-ICS, Analysis of the cyber attack on the Ukrainian power grid (2016). https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf. Accessed 03 Jan. 2021
47. D. Shelar, S. Amin, Security assessment of electricity distribution networks under DER node compromises. IEEE Trans. Control of Netw. Syst. **4**(1):23–36 (2017)
48. K. Smith, I. Wilson, The challenges of the internet of things considering industrial control systems, in *Privacy, Security And Forensics in The Internet of Things (IoT)*, ed. by R. Montasari, F. Carroll, I. Mitchell, S. Hara, R. Bolton-King (Springer International Publishing, Cham, 2022), pp. 77–94. https://doi.org/10.1007/978-3-030-91218-5_4
49. G. Spanos, A. Sioziou, L. Angelis, WIVSS: a new methodology for scoring information systems vulnerabilities, in *Proceedings of the 17th Panhellenic Conference on Informatics. PCI '13*

(Association for Computing Machinery, Thessaloniki, Greece, 2013), pp. 83–90. https://doi.org/10.1145/2491845.2491871

50. Tenable-community, Nessus (2021). https://www.tenable.com/products/nessus. Accessed 13 Oct. 2021
51. V. Visoottiviseth, P. Akarasiriwong, S. Chaiyasart, S. Chotivatunyu, PENTOS: penetration testing tool for internet of thing devices, in *TENCON 2017—2017 IEEE Region 10 Conference* (2017), pp. 2279–2284. https://doi.org/10.1109/TENCON.2017.8228241
52. D. Votipka, R. Stevens, E. Redmiles, J. Hu, M. Mazurek, Hackers versus testers: a comparison of software vulnerability discovery processes, in *2018 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 374–391. https://doi.org/10.1109/SP.2018.00003
53. B. Wang, X. Li, L.P. de Aguiar, D.S. Menasche, Z. Shafiq, Characterizing and modeling patching practices of industrial control systems. Proc. ACM Meas. Anal. Comput. Syst. **1**(1). https://doi.org/10.1145/3084455
54. S.A. Weed, US policy response to cyber attack on SCADA systems supporting critical national infrastructure (2017). https://media.defense.gov/2017/Nov/20/2001846609/-1/-1/0/CPP0007_WEED_SCADA.PDF. Accessed 02 Mar. 2022
55. Y.S. Wu, B. Foo, Y. Mei, S. Bagchi, Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS, in *Proceedings of the 19th Annual Computer Security Applications Conference. ACSAC '03* (IEEE Computer Society, USA, 2003), p. 234
56. Q. Xueqiu, S.W. Jia, C. Xia, L. Lv, Automatic generation algorithm of penetration graph in penetration testing, in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (2014), pp. 531–537. https://doi.org/10.1109/3PGCIC.2014.104
57. G. Yadav, P. Gauravaram, A.K. Jindal, SmartPatch: a patch prioritization framework for SCADA chain in smart grid, in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. MobiCom '20* (Association for Computing Machinery, London, United Kingdom, 2020). https://doi.org/10.1145/3372224.3418162
58. G. Yadav, P. Gauravaram, A.K. Jindal, K. Paul, SmartPatch: a patch prioritization framework. Comput. Ind. **137**, 103595 (2022). https://doi.org/10.1016/j.compind.2021.103595. https://www.sciencedirect.com/science/article/pii/S0166361521002025
59. G. Yadav, K. Paul, PatchRank: ordering updates for SCADA systems, in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (IEEE ETFA)* (2022). https://doi.org/10.1109/ETFA.2019.8869110
60. G. Yadav, K. Paul, Architecture and security of SCADA systems: a review. Int. J. Critic. Infrastr. Protect. **34**, 100433 (2021). https://doi.org/10.1016/j.ijcip.2021.100433. https://www.sciencedirect.com/science/article/pii/S1874548221000251
61. G. Yadav, K. Paul, Global monitor using spatiotemporally correlated local monitors, in *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)* (2021), pp. 1–10. https://doi.org/10.1109/NCA53618.2021.9685330
62. G. Yadav, K. Paul, A. Allakany, K. Okamura, IoT-PEN: a penetration testing framework for IoT, in *2020 International Conference on Information Networking (ICOIN)* (2020a), pp. 196–201. https://doi.org/10.1109/ICOIN48656.2020.9016445
63. G. Yadav, K. Paul, A. Allakany, K. Okamura, IoT-PEN: an E2E penetration testing framework for IoT. J. Inf. Process. **28**, 633–642 (2020b). https://doi.org/10.2197/ipsjjip.28.633.
64. T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, E. Kirda, Beehive: large-scale log analysis for detecting suspicious activity in enterprise networks, in *Proceedings of the 29th Annual Computer Security Applications Conference. ACSAC '13* (Association for Computing Machinery, New Orleans, Louisiana, USA, 2013), pp. 199–208
65. X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, D. Zhang, Robust log-based anomaly detection on unstable log data, in *ESEC/FSE 2019. Tallinn, Estonia: Association for Computing Machinery* (2019), pp. 807–817. https://doi.org/10.1145/3338906.3338931

# Review of Cyber Security for Power Trading and Communication Systems

**Aklilu Daniel Tesfamicael, Vicky Liu, and Matthew McKague**

**Abstract** The trading and communication systems of the wholesale energy market are an essential part of critical national infrastructure. If adversaries were to exploit the vulnerabilities in the wholesale energy trading and communication system, they could disrupt electricity generation and supply nationally, resulting in a devastating chain reaction. In this context, this study provides a review of deployments of security mechanisms for energy market trading and communication systems. This helps to understand the current security controls and challenges better and shines a light on potential research that can be conducted to make trading and communication systems more secure. This review is categorised into four themes: (1) security technologies that can be applied to energy trading and call audit systems, (2) blockchain technology that can be applied to protect energy trading and auditing services, (3) communication technology (voice over IP and video conferencing) that operates in the cloud, and (4) network performance and security management for voice over IP and video conferencing systems. This review investigates the use of blockchain technology that has increasingly emerged in a microgrid (peer-to-peer) energy trading and reveals a gap in using blockchain for macrogrid national energy trading. This study also emphasises the importance of balancing network security and performance when systems are hosted in the cloud.

**Keywords** Blockchain · Cyber security · Cryptocurrency · Power trading security · Security and performance · Security technology · Network security

A. D. Tesfamicael (✉) · V. Liu · M. McKague
Queensland University of Technology, Brisbane, QLD 4000, Australia
e-mail: aklilu.tesfamicael@qut.edu.au

V. Liu
e-mail: v.liu@qut.edu.au

M. McKague
e-mail: matthew.mckague@qut.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_4

## 1 Introduction

This chapter aims to provide a critical review of existing literature relevant to the deployment of security mechanisms for trading and communication systems, mainly for the trading and communication systems for the macrogrid wholesale energy market. The difference between microgrid and macrogrid is that microgrid is a group of decentralised (localised) energy micro-generators that act as a single controllable or integrated entity with respect to the grid. However, macrogrid is the traditional centralised grid where large generators (power plants) dispatch electricity through a national power transmission and distribution network. The centralised power generators mainly operate on fossil fuels, are integrated into the grid system, dispatching electricity nationwide.

This review is based on the growing acknowledgement that energy trading and communication systems are essential for critical national infrastructure. Moreover, cybersecurity, crucial to national security, has become an essential protection paradigm for such trading and communication systems.

This review has four themes that are structured around concepts. The first theme examines the security technologies relevant to power trading and communication. The second theme examines data security in trading systems and auditing services based on the application of blockchain. The third theme investigates communication technology and platform based on cloud-based voice and video conferencing. The fourth theme investigates the network performance and security management aspects of voice and video conferencing services. Finally, the conclusion and future work of this review are presented.

## 2 Theme 1: Security Technology

This section reviews security technology that can be applied to energy trading and call audit systems, including blockchain technology, consensus mechanisms, smart contracts, and virtual private networks (VPNs).

### Blockchain Technology

Nakamoto [1] introduced Bitcoin as a decentralised currency using blockchain technology. Blockchain technology is a shared and distributed data structure (ledger) across a network. Its data structure is formed by a time sequence of digital blocks of transactions without the need for a central authority. Blockchain is an irreversible distributed ledger upon which applications can be built [2]. Each participant in the blockchain network has a complete copy of the ledger. Everybody agrees on which transactions have been recorded and in what order.

Blockchain technology can be categorised into three types: public blockchain, private blockchain, and consortium blockchain. These types of blockchains differ in several ways, affecting the level of security they can provide. The main difference is that anyone can join the blockchain network in a public permissionless blockchain at any time, whereas the private and consortium blockchains are based on preselected membership. Private and consortium blockchains are both permissioned blockchains. In a private blockchain, the control of authorisation falls under one entity, while in a consortium blockchain, it is under the management of a group rather than a single entity. In a private or consortium permissioned blockchain, only authorised users can add entries into the blockchain system. Trusted parties are preselected to allow access to the blockchain system.

Bitcoin, Ethereum, and EOS are ranked in the top three public blockchains [3]. Buterin [4] introduced Ethereum in 2013. Ethereum is an open-source, decentralised network platform with smart contract functionality. Ethereum provides its own digital currency, Ether (ETH), similar to BTC of Bitcoin's digital currency. Developers can use Ethereum as a platform to create digital currencies and run decentralised applications (DApps) and smart contracts without the need for central control. Ethereum operates in a permissionless public blockchain environment and uses proof-of-work (PoW).

One example of a consortium blockchain is the Hyperledger Fabric [5], consisting of peer nodes that hold copies of ledgers and copies of smart contracts. Hyperledger Fabric is a permission-based blockchain which has been widely studied in academia and is applied in industrial fields [5–9]. One of the driving reasons behind the increasing adoption of this technology is its enhanced architecture and improved throughput over the other types of blockchain [10, 11].

### Security Properties of Blockchain Technology

The main security properties of blockchain technology are: (1) all participants agree on which transactions are stored in the ledger and the order in which they appear. It is difficult for an attacker to disrupt this condition [12], (2) it is challenging for an attacker to change or remove a transaction or to insert a transaction (other than appending a transaction in the usual way) and (3) it is very difficult for an attacker to prevent an honest transaction from being appended to the ledger [13, 14]. In the case of the trusted third party (TTP), TTP keeps a list of transactions. A blockchain member (participant) can ask TTP to add a transaction, and the TTP will check that it meets any application-specific conditions (for example. sufficient funds and appropriate permissions) and add it to the end of the list. A participant can ask to see the ledger, in which case the TTP sends the participant the list of all transactions (or whichever ones the participant requested.)

Blockchain technology also builds on providing hash chained data storage, digital signature, consensus, and authorisation.

The hash serves as evidence of the transaction and is used for state validation and audit purposes [15]. The hash is for the previous block in the blockchain so that data in all blocks are connected together in a chain from the initial block to the most recent block. Any attempt to delete or change a block will break the chain of hashes and be detectable.

Digital signatures are a fundamental building block in blockchains; they are primarily used to verify the authenticity of transactions. It is also a scheme for verifying that data has not been tampered with and the origin of the data. Digital signatures link an identity to a particular message using cryptographic techniques and are difficult to forge due to their use of public-key cryptography. Users own both a public key and a private key, forming a pair. Digital signatures are primarily used in the blockchain for the authenticity of the transaction. When users submit transactions in the blockchain system, they prove to all nodes that they are authorised to submit transactions.

Authentication and transaction integrity are essential concepts in a permissioned blockchain. In the case of Hyperledger Fabric, participants in the blockchain network have their own two keys assigned to them. Transaction proposals are digitally signed using an owner's private key, which also includes a public key in the transaction payload sent to peers and orderers. Peers and orderers then verify the signature using the owner's public key [5].

The consensus mechanism ensures that all the nodes agree on what transactions are on the chain and the order of transactions. All nodes validate the information to be appended to the blockchain, and a consensus protocol ensures that the nodes agree on a unique order in which entries are appended [16].

Permissioned blockchains are built based on authorisation and authentication as a requirement to login into the blockchain system. Only authorised parties are allowed to submit and/or access transactions in the blockchain system. If data are sent from an unauthorised user, the signature of that user will not be matched, and the data will not appear on the chain. A permissioned blockchain uses an authorisation layer that determines the blockchain members and allows them access to the system [17].

### Application of Blockchain Technology

Permission-based blockchain technology supports different security properties. Putz et al. [18] utilised permissioned blockchain to preserve the integrity of log records, a technique that does not depend upon a trusted third party.

Gai et al. [17] developed a permissioned blockchain model for a peer-to-peer network to address privacy and energy security issues. They implemented smart contracts on the permissioned blockchain with an optimal security-aware strategy.

Several authors explored using permissioned blockchains in a variety of applications. Hyla and Pejaś [19] explored using blockchain for security provisions to protect medical records. Hyla and Pejaś utilised permission-based blockchain with the BFT consensus protocol to store sensitive medical records. Hyla and Pejaś applied permission-based blockchain to manage access control over medical records. Some studies [20, 21] demonstrated the application of the permission-based blockchain for integrity control, record access management, and user authentication for healthcare-related systems. Numerous studies [22–26] also applied permission-based blockchain to preserve privacy and verify the vote results of the electronic voting systems.

Brotsis et al. [27] applied permissioned blockchain in the (IoT) to enhance security and evaluate performance and fault tolerance. They assessed the performance and tolerance of permission-based blockchain with different consensus mechanisms in an IoT environment. Brotsis et al. contended it is advantageous to integrate the

Byzantine Fault Tolerance (BFT) protocol into Hyperledger Fabric for IoT. The use of blockchain technology as a secure access control mechanism for IoT was also studied by Pal et al. [28]. Pal et al. employed an Ethereum-based blockchain to authenticate the participant's identity. Some studies [29, 30] addressed privacy-preserving methods and security challenges when blockchain is applied to IoT applications.

### Consensus Mechanism

In the blockchain network, consensus refers to reaching a common agreement on the content of the distributed ledger. There are various consensus mechanisms used in the blockchain. Proof of work (PoW) and Proof of Stake (PoS) are the most common consensus mechanisms for permissionless blockchain, while the Practical Byzantine Fault Tolerance (PBFT) algorithm only works in permissioned blockchain. Typically, consensus mechanisms are useful for private or public blockchain, but not both. The common consensus mechanisms used for public blockchain are PoW, PoS, and PoA, and the common ones for private is PBFT.

Numerous studies [31–34] employ PoW and PoS mechanisms for a permissionless public blockchain. Proof of Authority (PoA) is a reputation-based consensus algorithm that leverages the value of identities. PoA is usually used for public blockchains where anyone can contribute transactions or read the ledger. PBFT performs better than PoA, where data integrity is essential [35]. PoW, PoS, and PoA could also be used for private blockchains, but since PBFT offers much better security, cost tradeoff, and speed, they are not advancing into practical implementation for private blockchains.

PoW is one of the widely adopted methods in blockchain and was popularised by Bitcoin. The main working principle of PoW is to leverage computing with a massive amount of computational processing power and time to solve complex mathematical equations. The algorithms are used to confirm transactions and produce new blocks to the chain. A new block can only be created if the cryptographic hash value of the last recorded block is ascertained by solving a complex equation. Peers participating in the consensus mechanism (namely miners) compete to be the first to solve the problem. The first miner to submit their block (namely a set of transactions) and the solution gets a reward, and the block is added. If there is more than one valid block, then the tiebreaker is whichever one has the longest chain of blocks coming after it.

In order to subvert the mechanism, a dishonest party needs to consistently be the first miner or create a chain longer than that the honest parties are building. Both of these require having more computing power than the honest parties (namely > 50% of the total). A malicious attack could occur if the user or a group of users collectively control more than 50% of the computing power on the blockchain network. If under attack, the malicious nodes could broadcast a fraudulent transaction to rewrite all historical transactions. PoW may come at a high cost as it requires enormous resources (such as power) to process complex equations [36].

PoS is an alternative to PoW to avoid high resource consumption issues [37]. In PoS, miners (users) do not require immense computing power for the competition process. No competition is required to determine the next transaction block, as the blocks are created by peers (provers), which are chosen randomly based on the fraction of currency they own. PoS could process transactions and generate blocks much faster than the PoW [38–40]. In order to add a malicious block, one would have to own 51% of all the digital currency on the blockchain network. The cost of owning an immense amount of coins in PoS is higher than the cost of computational power to solve complex mathematical equations in PoW [41]. PoS have been explored since 2015 to realise its benefits, which is more efficient than PoW [31–33, 42].

A consortium blockchain is a semi-private system and has a controlled user group but works across different organisations. For consortium blockchain, a PBFT can provide adequate solutions [43, 44]. With PBFT, each blockchain node needs to know the identity of every other blockchain node on the network. Consensus in the PBFT can be reached when the number of Byzantine faults (malicious nodes) is less than one-third of the total number of nodes [45]. This is enabled by the fact that all honest nodes agree on the system state at a specific time. In the PBFT algorithm, trusted blockchain members are predefined, and the algorithm is efficient to provide high transactional speed, low energy consumption, and system cost. PBFT provides transaction finality without the need for additional confirmations once approved, and as it is not computationally intensive, a substantial reduction in power consumption is achievable compared to PoW.

Additionally, permissioned-based blockchains are private and are by invitation with known identities. The limit on identities is helpful because it prevents Sybil attacks, namely an attacker masquerading multiple people. If an attacker could create identities at will, they can create enough to control more than the 1/3 threshold required to break the consensus mechanism. Vukolić [36] proposed PBFT protocols, including the PBFT consensus algorithm supported by Hyperledger Fabric. Table 1 lists the comparison of the common consensus mechanisms.

### Smart Contract

A smart contract is a computer program stored on a blockchain that is automatically executed when predetermined terms and conditions of a contract are fulfilled, without the need for intermediaries. Smart contracting was initially introduced in 1994 by Szabo [46]. Szabo defined a smart contract as a computerised transaction protocol that executes the terms of a contract.

**Table 1** Comparison of consensus mechanism

| | PoW | PoS | PBFT | PoA |
|---|---|---|---|---|
| Type of blockchain | Permissionless (Public) | Permissionless (Public) | Permissioned | Permissionless permissioned |
| Performance | Low | High | High | High |
| Adversary tolerance | Less than 50% | Less than 50% | Less than 33.3% | 50% |
| Energy saving | No | Yes | Yes | Yes |
| Example | Bitcoin, ethereum | Peercoin | Hyperledger fabric | Ethereum |

An example of the application of a smart contract is escrow. A smart contract could be used to implement an escrow that releases funds after all the transaction terms are met. Escrow is a legal concept in which a trusted third party holds a fund on behalf of two other parties processing a transaction [47, 48]. The fund is on hold until contractual obligations have been satisfied. An escrow is for two parties that are not trusting each other and relying on a trusted third party. This is an example of using smart contracts in a case where there is no mutual trust between parties, thus eliminating the need for a trusted third party.

*Applications of Smart Contract*

Dai et al. [133] and Cohn et al. [49] explored the applications of smart contacts in various cases to reduce administrative overhead and human errors, including banking and financial service contracts, trading processes, and insurance contacts. Smart contracts can also be applied to peer-to-peer energy trading services to trade energy with no central entity [50, 51].

One example of a blockchain-based form of finance is decentralised finance (DeFi). DeFi is built on top of the Ethereum blockchain that does not require a central financial institution. DeFi applies smart contracts to automatically execute agreements without requiring intermediaries in an open and transparent way [52].

Ellul and Pace [53] investigated applications of smart contracts in the IoT environment. Ellul and Pace developed a split-virtual machine, AlkylVM, to create a smart contract to interact with the resource-constrained IoT devices to communicate with the blockchain system.

Several studies [54, 55] examined applying smart contracts to agree on a purchase and sale of property. This is to streamline the processes and reduce costs for rental agreements.

Depending on the environment and the type of node, blockchain technology can be divided into three types, public blockchain, private blockchain, and consortium blockchain. Each consists of six basic layers: the data layer, network layer, consensus layer, incentive layer, contract layer, and application layer, as listed in Table 2.

**Table 2** Layers of blockchain technology

| Blockchain layers | Principal components |
|---|---|
| Application layer | Cryptographic currency |
| Contract layer | Smart contracts, algorithmic mechanism |
| Incentive layer | Distribution mechanism, smart contract, script codes |
| Consensus layer | Proof-of-work (PoW), proof-of-stake (PoS), practical Byzantine fault-tolerant (PBFT) |
| Network layer | Peer-to-peer network, Propagation mechanisms |
| Data layer | Data block, chain structure, time stamp, hash function |
| Application layer | Cryptographic currency |
| Contract layer | Smart contracts, algorithmic mechanism |

### Authentication Mechanism

Asymmetric-key cryptography is commonly referred to as "public-key cryptography". For encryption and authentication purposes, it uses a mathematically associated key pair—a public key and a private key. A digital certificate is used to attest to the binding between a particular entity and its public key by a trusted third party, known as a Certification Authority (CA), under the Public Key Infrastructure (PKI) scheme.

Regarding authentication, numerous studies focus on certificate-based authentication for blockchain networks. Fromknecht et al. [56] proposed a decentralised PKI called Certcoin. Fromknecht et al. contended that Certcoin could provide identity retention, but it lacks the practical experimentation to assess the feasibility. Studies [56–64] focused on adopting distributed PKI for blockchain but lacked a practical performance assessment in real-time applications. Numerous studies [68–74] successfully implemented systems based on the decentralised PKI.

Javaid and Sikdar [65] proposed a framework that uses blockchain and PKI with a dynamic Proof-of-Work (dPoW) consensus for secure grid energy trading. With PKI, an authentication mechanism for an electric vehicle is established.

Pallickara and Fox [66] investigated a P2P grid that comprises grids and P2P networks services, whereby authentication mechanism is plugged into the P2P system to authenticate users to access the system.

Digital certificates and PKIs are used to provide an authentication mechanism [67–70].

### Virtual Private Network (VPN)

A virtual private network (VPN) is a network security technology used for data confidentiality and integrity when transferring sensitive data over public networks. VPNs can be used to mitigate network-based attacks such as man-in-the-middle, IP spoofing and port scanning attacks. There are different types of VPN. One of the commonly used types is Internet Protocol Security (IPsec), developed by the Internet Engineering Task Force (IETF). IPSec can be configured to operate in either transport or tunnel modes. Transport mode can provide data security to IP payload (data), whereas Tunnel mode can protect the entire IP packet containing the IP header and payload. Tunnel mode is achieved by encapsulating the original IP packet in another new IP packet, a process which is referred to as IP tunnelling [71, 72].

Liu et al. [71] state that the key components of IPsec implementation include Authentication Header (AH) Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE). AH can be used to provide authentication for data integrity against replay attacks. ESP can be implemented to support data confidentiality and integrity against replay and packet sniffing attacks. AH and ESP can be used alone or together to provide trusted transmission of sensitive data.

Numerous studies [73–77] have referenced the use of IPSec or alternative security measures to protect VoIP services and analysed the performance factors when implementing IPSec to carry VoIP traffic. Kuhn et al. [144] proposed to use IPSec tunnelling with ESP mode to protect communications between the callee and the caller. Thanthry et al. [78] and Choudhury [79] proposed an encryption scheme that uses a public key scheme for authentication and encrypts the voice traffic with a symmetric cipher scheme.

Liu et al. [71] evaluated security performance when IPSec is implemented in various IPSec types on different operating systems. They provided insights on controlling performance metrics when designing and managing network security for other operating systems. They argued that the appropriate balance between network security and performance could be achieved by applying the metrics for network security and overall performance when implementing IPSec VPNs.

*Quality of Service (QoS) for UC.*

Quality of service (QoS) is the measurement of the overall performance of a service, such as voice or video conferencing. QoS is one of the critical success factors for the deployment and performance management of unified communications systems, particularly for cloud-based unified communications systems. When implementing unified communications over IPSec, it is crucial to monitor the performance of IPSec VPN to avoid service quality degradation. The delay, jitter, and packet loss affect the QoS of the unified communications. Therefore, applying security measures to time-sensitive applications must be balanced with performance. Nowadays, VPNs are a commonly used network security mechanism over IP-based networks. IPSec VPNs can be implemented in tunnel mode to protect data in transit.

The ITU Telecommunication Standardization Sector (ITU-T) G.114 recommends that a set of QoS requirements for a real-time communications system be satisfied for the voice services to be of high quality, as shown in Table 3.

### Summary

This section reviewed security technology that can be applied to energy trading and call audit systems. We have observed a lack of study conducted on applying security to UCaaS, particularly UCaaS over IPSec, and the deployment of UCaaS in public or private clouds. UCaaS can be regarded as a cost-effective model for the on-demand delivery of unified communications services in the cloud. However, addressing security concerns has been seen as the biggest challenge to the adoption of UC in the cloud. Though many businesses have shown great interest in UCaaS solutions, much work is to be done to protect UCaaS services from security threats.

**Table 3** Voice over IP QoS requirement

| QoS parameters | ITU-T G.114 recommendation |
|---|---|
| Jitter | Less than 30 ms |
| Latency | 150 ms or less (one-way end-to-end) |
| Packet loss | Less than 1% |
| Mean opinion score (MOS) | 4 or above |

Undoubtedly, the UC traffic is more sensitive to latency, jitter, and packet loss than most network applications. Quality of service (QoS) is one of the critical success factors in UC services' design and performance management. When implementing UC over IPSec, it is also crucial to analyse its performance to avoid service quality degradation that the overhead of IPSec may add.

## 3 Theme 2: Security Technology for Protecting Energy Trading and Auditing Systems

A secure energy trading system is essential for energy generators, energy market regulators, and customers to buy and sell electricity in the energy market. This section reviews blockchain technology that can be used to protect energy trading and auditing services.

The current Australian energy market trading system is shown in Fig. 1. The Australian Energy Market Operator (AEMO) manages the electricity markets across Australia.

### Application of Blockchain Technology Applied to Secure Energy Trading

Since 2017 we have seen a growing interest in blockchain applications in the energy sector. Researchers have developed a blockchain-based application for the energy sector, focusing on peer-to-peer solar energy trading, renewable energy certificates (REC), grid management, carbon credits tracking, energy storage, and electric vehicle charging. However, this review focuses on applying blockchain technology in energy trading and security; as such, using blockchain schemes in REC, grid management, carbon credits tracking, energy storage, and electric vehicle charging is outside the scope of this review.

Numerous studies have utilised blockchain technology to address the security of energy trading. In 2019, Andoni et al. [80] reviewed 140 blockchain-focused studies. The main focus of their research was to map the potential and relevance of blockchain applications to the energy sector. These reviews provide insights into the security challenges the energy sectors face because the energy systems are undergoing rapid changes to accommodate the increasing volumes of embedded renewable generation. In addition, they looked at blockchain technology's opportunities, challenges, and limitations in several cases, including peer-to-peer energy trading, electric vehicle charging, and decentralised energy markets. These provide a step forward in identifying where blockchain can be applied to enhance security, management, and payments in energy market-related trading.

Along with use cases in various sectors, the potential of blockchains in the energy industry has just started to evolve, as shown by the increasing number of research projects, start-ups, pilots, and trials [34, 81, 82]. Besides, some consulting firms, such as Deloitte [83] and PwC [84], reported the possibility of blockchain technology disrupting the energy sector by using cryptocurrencies for trading energy. Deloitte

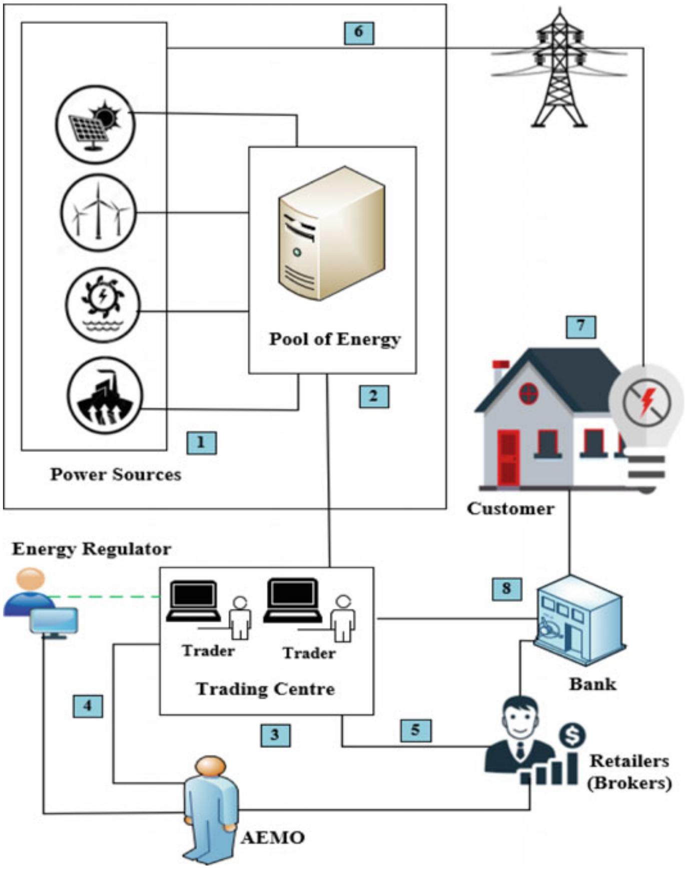


Fig. 1 Current energy market trading architecture

[83] investigated technical, cultural, organisational, and commercial challenges in adopting blockchain technology. PwC [84] explored the challenges and opportunities of blockchain technology for prosumers (namely, households that consume, produce, and sell energy). However, most studies on blockchain applications are still in the pilot and early deployment stages.

Aitzhan and Svetinovic [85] used blockchain technology to address security issues in peer-to-peer energy trading. They developed a token-based energy trading system called PriWatt and claimed to trade energy securely in a peer-to-peer network. They use a case study to evaluate system performance and analyse security. To reach a consensus, they applied a proof-of-work (PoW) mechanism. They have also used multi-signature and anonymous encrypted messaging to enable peer nodes to anonymously negotiate energy prices and secure trading transactions.

In a similar concept, Dimitriou and Karame [86] focused on protecting user privacy concerning billing and trading transactions for peer-to-peer energy trading in a microgrid. Their research focused on securing transactional information of energy consumers (customers) in the smart grid distribution network. Their study describes a method for securely anonymising frequent (for example, every few minutes) electrical metering data sent by a smart meter. More studies [87–89] have also addressed security and privacy for energy trading in microgrids.

Li et al. [90] discussed the energy trading scheme's common security and privacy challenges using Industrial Internet of things (IIoT) technology for peer-to-peer network connectivity. The authors proposed a secure energy trading system using consortium blockchain technology without a trusted third party. They showed that the credit-based payment method of the blockchain system improves the security and efficiency of peer-to-peer networks and supports fast peer-to-peer energy trading services.

Hassan et al. [91] developed a consortium blockchain-based approach for microgrid energy auction in a peer-to-peer network. According to the authors, differential privacy techniques (a technique for publicly sharing information without revealing information about the individual) can make an auction more secure and private. They compared their approach with Vickrey-Clarke-Groves (VCG) auction scenario [92] using an experimental analysis, and their result showed higher security than that of the VCG mechanism.

A number of studies [93–104] examined using blockchain technology for energy trading in the microgrid network. Paudel and Beng [94] proposed a hierarchical peer-to-peer (HP2P) energy trading framework designed to service a designated community in a microgrid distribution network. Based on the experiment in a microgrid system, Paudel and Beng argued that their proposal could reduce operational costs. Similarly, Xie et al. [101] proposed a conceptual framework between prosumers for trading energy in a microgrid peer-to-peer energy trading that is not related to the macro grid.

Nunna et al. [98] proposed an agent-based energy management system simulation with energy storage systems for microgrid-based energy trading. Nunna et al. developed bidding mechanisms with energy storage functions for prosumers to trade in the energy market. Their simulation results indicated that the participation of prosumers in the energy market through energy storage systems might reduce electricity consumption costs.

Only a few studies focus on the practical implementation of energy trading systems. Tushar et al. [105] explored the challenges in peer-to-peer energy trading using blockchain, including the billing complexity, lack of practical implementation, the limitation of feeding excess energy into the grid, and data privacy when trading energy in the peer-to-peer network. Kwak and Lee [99] simulated peer-to-peer energy trading using the Ethereum-based blockchain. Such a study is a step forward in providing practical blockchain technology implementations for its real-world deployments.

*Privacy and Anonymity in Blockchain—Mixing protocols*

A growing number of studies recognise the importance of blockchain technology in securing transactions using mixing or cryptocurrency techniques. Since the introduction of Bitcoin in 2008 by Nakamoto [1], numerous cryptocurrencies have been developed. Many solutions have been proposed to improve the anonymity of Bitcoin and cryptocurrencies in general. Some of the proposals are based on mixing protocols (mixing coins), while others focus mainly on cryptographic techniques [106].

Bitcoin mixers or tumblers are services that mix digital coins with other coins to secure privacy. The Mixnet structure is one way to achieve anonymity. This method relies on a trusted third party to combine transactions. Users can deposit their funds with a third party to combine digital currency into a single transaction and send them to the intended recipient. The important part is that many users simultaneously submit their funds to the TTP, who then forwards payments to the recipients. It is hard to know which sender goes to which recipient.

The purpose of breaking the links between input–output addresses is that the transaction’s input and output addresses cannot be linked together, thus preserving anonymity. Link obscuring mechanisms are added to the middle of a transaction to break links to avoid linking transactions to the real identity.

Various privacy and anonymity improvement techniques have been investigated that use the mixing techniques: hiding amounts for transactions to improve privacy, concealing the source and destination IP addresses of a user, breaking the links between input–output addresses of a transaction, and breaking links between transactions [107]. The most evolving mixing protocols that have been discussed in the literature are: Mixcoin, Blindcoin, Bitcoin tumblers, CoinShuffle, CoinShuffle++ and Möbius.

Bonneau et al. [106] proposed a mixer mechanism to facilitate anonymous payments in Bitcoin. It allows users to combine their digital currency with other users to preserve their privacy. However, using Mixcoin can link to private user information, leads to compromising user privacy. To address the privacy issue of the “Mixcoin”, Valenta and Rowa [108] proposed another digital currency called Blindcoin that modifies the Mixcoin protocol. Blindcoin, unlike the Mixcoin, hides the user’s input and output address of the transaction from the mixing server (a third-party server for achieving anonymity) to preserve user privacy. Blind signatures ensure that the Mixer cannot link the input and output addresses. However, the privacy-preserving methods such as that of Blindcoin come with drawbacks. The main downsides of Blindcoin are the loss of mix indistinguishability and the necessity to maintain two unlinkable identities A and A’ [108].

Some mixing protocols require intermediaries to process payments to preserve users’ anonymity, such as the TumbleBit mechanism proposed by Heilman et al. [109]. The TumbleBit is an anonymous payment protocol that allows users to make payments through an untrusted intermediary, Tumbler T, to achieve anonymity. In contrast, Ruffing et al. [110] proposed the CoinShuffle Bitcoin mixing protocol that does not require any third party (intermediary), unlike TumbleBit, which allows users to utilise Bitcoin anonymously. Ruffing et al. [111] presented a CoinShuffle++, a mixing protocol for Bitcoin users that is improved from CoinShuffle. Ruffing et al. demonstrated CoinShuffle++ to perform better than its predecessor, CoinShuffle. The analysis results show that in a scenario with 50 participants, a transaction can be created in eight seconds compared to Coinshuffle, which requires almost three minutes.

Interestingly, some other studies proposed to store financial transactions outside the blockchain system, such as the Hawk method proposed by Kosba et al. [112]. The Hawk system is a decentralised smart contract system that can operate from an external source to the blockchain system so that financial transactions are not stored within the blockchain system. Hawk is different from the other protocols discussed above as it does not use any mixing techniques and can still maintain users' privacy. Another privacy protocol, Möbius, an Ethereum-based tumbler or mixing service proposed by Meiklejohn and Mercer [113], demonstrated its operation in the Ethereum-based platform. Möbius, achieves a much lower off-chain communication complexity than other existing tumblers. Senders and recipients are required to send only two initial messages in order to engage in a transaction. One of the benefits of Möbius, Meiklejohn and Mercer argued is its better privacy functionality as malicious senders cannot identify which pseudonyms belong to the recipients to whom they sent money.

As evident from the literature above, some limitations exist in the adaptation of the mixing protocols. The key limitation is the need for trusted third-party mixing services. Larger transaction processing time is also another factor that limits mixing protocols from their practical implementation. Additionally, mixing protocols have not achieved an acceptable level of confidentiality though the unlinkability of the transaction message has improved slightly recently.

*Privacy and Anonymity in Blockchain—Cryptographic techniques*

The most evolving cryptographic solutions discussed in the literature are: Zcash, Monero, Aztec, Zether, ZeroCash, and Zerocoin.

Zcash is believed to be one of the more robust privacy and anonymity guarantees [114]. It uses a zero-knowledge proof method to present the validity of a transaction without revealing any personal information. It enables privacy features for transactions so that sender and receiver addresses, and the amount of money transacted, can be kept private. The addresses beginning with a "t" (t-addrs) are considered transparent and are similar to bitcoin transactions. "Shielded" transactions are used with addresses starting with a "z" (z-addrs), and these are entirely anonymous. Numerous studies [115–118] adopt Zcash to provide privacy and anonymity features for blockchain.

Sun et al. [119] presented the necessary properties and security requirements of Ring Confidential Transaction (RingCT) for cryptocurrency Monero. Monero achieves anonymity through ring signatures, stealth addresses, and Ring CT. Ring signatures allow a single user out of a set of users to sign a message, but it is not possible to determine which user in the set made the signature.

There is a lack of study on the evaluation of how privacy can be preserved when executing smart contracts in Ethereum [120] as Zcash, Monero, and Mixnets do not provide smart contracts. In general, a lack of sophisticated privacy preservation techniques exists in Ethereum. All transactions are public that can reveal personally identifiable information.

AZTEC [121] and Zether [120] are payment mechanisms that are compatible with Ethereum and other smart contract platforms. Both mechanisms are based on smart contracts, so the expensive consumption of the pricing value to successfully conduct a transaction (GAS) prevents it from practical use [122]. AZTEC and Zether can incorporate the Ethereum-based blockchain to preserve privacy and anonymity.

Rondelet and Zajac [120] examined the Ethereum network properties for protocol 20 (ERC20) digital currency. They proposed the digital currency ZETH, and claim that it allows secure and private payments operating on the Ethereum network.

Somin et al. [123] demonstrated that for preserving privacy, the AZTEC protocol is compatible with Ethereum-based blockchain with smart contracts (such as Ethereum's public ERC-20 token) to convert it to a confidential AZTEC note.

To address the privacy issues of Bitcoin, numerous studies [114, 124–129] developed cryptocurrency techniques, such as Zerocash and Zerocoin cryptocurrency, for preserving privacy and anonymity.

Regarding preserving the privacy of energy trading transactions, Mihaylov et al. [130] and Mihaylov et al. [131] introduced NRGcoin, a digital currency for renewable energy trading based on the concept of a decentralised blockchain application. Their study is for peer-to-peer trading in which prosumers feed excess electricity into the main grid and trade with digital currency. The smart meter counts one NRGcoin for every 1-kilowatt hour (KWh) of renewable energy the prosumer solar system feedback into the main grid. NRGCoin is generated by injecting energy into the grid rather than spending energy on computational power. It is directly linked to the monetary value of energy rather than to any particular value.

Similarly, Laszka et al. [132] introduced Privacy-preserving Energy Transactions (PETra), which enables trading energy in a secure and verifiable manner, preserves prosumer privacy and allows distribution system operators to regulate trading. PETra is based on digital tokens to represent the quantity of energy generation and consumption. Both the Mihaylov et al. and Laszka et al. proposals are for energy trading in microgrids, not in macrogrids.

### Application of Blockchain Technology in Auditing Services

Blockchain technology has been applied to numerous business applications, such as peer-to-peer energy trading, product tracking, logistics management system, supply chain management, intellectual rights management, and digital payments. However, little research has investigated its use in accounting and auditing, although [133–156] investigated blockchain applications in finance. The studies [133–156] can improve the actual application in finance validated by experimentation to assess its implementation practicality. These studies require more work to advance to the execution phase, given that this technology has gained momentum only recently and still lags concerning practical adoption.

Blockchain technology has attracted broad interest also in the auditing area. It is anticipated that blockchain will continue to be adopted in diverse areas. Dai and Vasarhelyi [149] contended that blockchain technology could transform the accounting ecosystem and auditing practices. They explored blockchain technology with smart contracts applied to financial and accounting applications to validate the audit trail data. One of the key benefits discussed in this paper was the capability of blockchain technology to bring together all parties: managers, accountants, auditors, business partners, and investors to work collaboratively to verify transactions, cross-validation capabilities, and auditing. For accountants and auditors, smart contracts play a significant role. Dai and Vasarhelyi [149] stated that some accounting processes could be automated by encoding business rules or agreements into smart contracts. If smart contracts execute rules such as invoice reconciliation, the system can automatically review, verifiy, and process payments. Antipova [151] discussed the benefit of using blockchain technology for government auditors to strengthen their power to investigate and collect evidence.

Kwilinski [157] and Zhang et al. [158] claimed that blockchain could be used in accounting applications as an alternative approach to traditional auditing and accounting operations. The authors argue that the auditing process could be performed more efficiently and effectively through the automation of the manual audit process. Using blockchain with smart contracts can eliminate the involvement of a third party in auditing tasks and improve efficiency by reducing manual processes [137, 159].

Schmitz and Jana [134] investigated 76 publications and focused on 16 academic publications that apply blockchain technology for accounting and auditing applications. Schmitz and Jana argued that the existing audit processes that are labor-intensive and time-consuming. Auditors receive journal entries, spreadsheet files, and other documents in electronic and manual formats that require a significant amount of time to plan before performing auditing. This manual audit process may come at a high cost to the organisation hiring auditors [160–162]. However, blockchain technology can be used to address this problem in an effective and efficient manner [163, 164]. Smart contracts enable reconciliation capabilities to reduce the significant amount of time auditors use to plan for audit tasks, eventually lowering the cost of conducting audit activities. Blockchain technology can be applied to decrease operational costs and reduce the risk of human error when performing auditing [137, 165].

The opportunities and challenges of using blockchain with smart contracts for accounting and auditing require a thorough investigation. Numerous studies [149, 158] discussed the use of blockchain in accounting, focusing on how this technology could enable verifiable accounting processes and transparent accounting practices.

Deloitte developed a blockchain platform named Rubix to target four applications: financial reconciliation, audits, land registry, and loyalty points [166]. In 2017, Deloitte claimed that it had successfully performed a blockchain-based auditing application. KPMG International Limited, in partnership with Microsoft, provides its blockchain platform to support business applications based on blockchain technology [167].

Only a few studies in using blockchain for telephone call auditing. Kozloski et al. [168] proposed a call tracking method using blockchain. Kozloski et al. can track and maintain call conversations using a blockchain-based peer-to-peer network. Placing call recordings on the blockchain system can impact the performance of the system.

Figure 2 illustrates the overall process used by the current Telephone Call Recording system. For example, anyone can make and receive a telephone call without any authentication (1). Once a telephone call is placed, a telephone call is established in the Telephone Exchange system (3). Meanwhile, the telephone call is logged in the recording system (2a). For those who use a manual logbook method, telephone call information is logged in a manual logbook entry (2b). An auditor conducts auditing either by reviewing telephone call recordings from the system (4a) or the manual logbook entries (4b).

### Summary

In the current trading system, energy is often traded using insecure means of communication that rely heavily on the existence of mutual trust between the parties. These time-consuming processes are open to human error and could potentially result in a breach of data integrity and confidentiality, intentionally or unintentionally. Current practice thus constitutes a significant risk to energy market data and the integrity of the transactions that underpin the provision of electricity from generators to customers. We have also observed that energy trading systems are undergoing a rapid transformation due in part to an increasing demand for renewable energy sources to be integrated into the macro power grid.

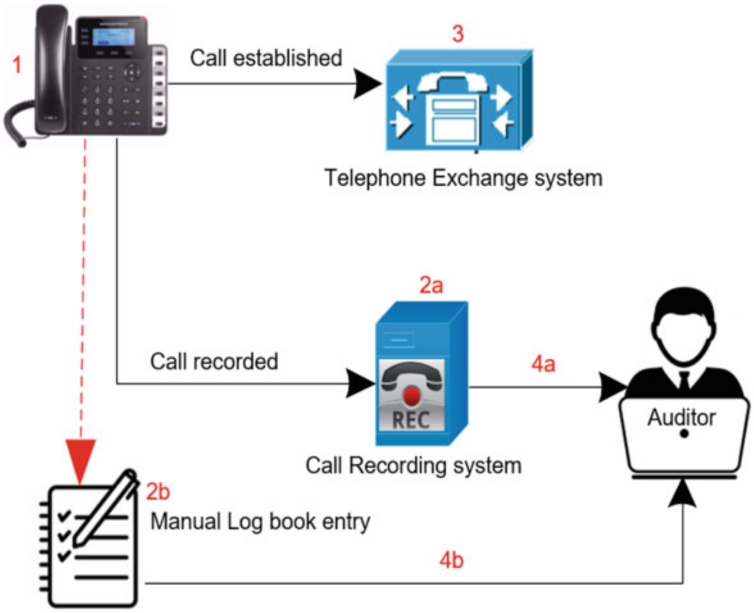


**Fig. 2** Current telephone call recording system

With increasing volumes of data being collected from macrogrids and energy participants, it is imperative to ensure the efficiency of operation, data integrity, and confidentiality. It is apparent that to guard against improper information modification, destruction, non-repudiation and authenticity of data are matters of vital concern to the integrity of electricity generation and trading.

The auditability of telephone call records also plays an essential governance role in the electricity industry. A breach of the integrity and confidentiality of telephone conversation records could have national security implications. Financial penalties and a tarnished reputation are further potential consequences of regulatory non-compliance. Currently, telephone call recording information is provided to auditors in various electronic and paper-based formats that require them to invest significant time when conducting telephone call recording auditing. In the absence of automatic verification and auditing capabilities, these recording processes are labour-intensive and prone to human error. The consequence is that the call recording process is vulnerable to activities that, intentionally or not, may breach the integrity of the recording of these conversations.

## 4 Theme 3: Communication Technology and Cloud Computing

This section reviews communication technologies and cloud platforms, focusing on Unified Communications (UC) and Cloud Platform Systems, particularly the security and performance aspects of cloud-based voice over IP (VoIP) and Video conferencing.

### Unified Communications

This section reviews the UC, its movement to the cloud environment, and the security mechanisms used to protect UC in the Cloud.

We have started to see the evolution of UC in the Information and Communication Technology (ICT) industry in recent years. Within ICT, a definition of UC is still debatable, but many define UC as the integration of real-time communication services or systems. For many, such communication services include "voice over internet protocol" (VoIP), video conferencing, instant messaging (IM), presence, and voicemail.

Recently UC has attracted much attention from major industry players and academic research communities [169–172]. UC integrates communication services, including instant messaging, voice over Internet Protocol (VoIP), and video conferencing. An increasing number of government agencies and business organisations are deploying UC for the first time or replacing their legacy telecommunications systems. UC can eliminate redundant facilities and converge all forms of communication systems under a single and integrated platform. By leveraging media convergence, UC can support effective communication in the workplace, resulting in increasing productivity and improving efficiency [173–175].

Lately, the cloud movement is driving UC from the on-premises model to a cloud model, namely Unified Communications as a service (UCaaS). UCaaS promises to be a cost-effective cloud-based UC service to meet on-demand services [175]. UC can be operated in three modes: on-premises, cloud or hybrid mode. The UC server hosted on the customer premises operates under the on-premise mode, while on cloud modes are those UC servers hosted in the Cloud.

Generally, traders in energy markets use a specialised trading phone. One of the widely used types of trading communication systems is the trading turret telephone, a special-purpose telephone communication system generally designed for the needs of financial traders and used at the trading desk. Most trading communication systems deployed on-premises are based on legacy circuit-switched and leased-line trading communication system architectures. Due to the gradual advancement of the Internet and related cost efficiency, it is increasingly attractive to move the trading communication systems to an IP-based solution with Session Initiation Protocol (SIP) connectivity. The evolution of this IP-based trading communication system solution advances the case for hosting a trading communication system in the Cloud. The IP-based trading communication system can integrate with other communication technologies to form a unified communication technology.

However, there is a lack of study about trading communication system security, performance, and legal requirements to ascertain if the requirements for trading communication systems in the Cloud can be met. Rehor et al. [176] discussed a trading communication system's regulatory and compliance requirements by specifying SIP-based media recording. They then discussed the signalling schemes security requirements for a trading communication system to protect from eavesdropping. Rehor et al. elucidated how critical the trading communication system must be to meet security and legal requirements.

Numerous studies [177–181] discussed an open-source automatic branch exchange (PBX) emulator, Asterisk, which is different from a trading communication system in its architecture and function. One of the Asterisk functions is to offer a converged circuit-switched and packet-based communications system. Such systems can also be hosted in the Cloud to offer hosted VoIP services. Asterisk can support a full UC service and a wide range of VoIP protocols, including H.323, the Media Gateway Control Protocol (MGCP), and Session Initiation Protocol (SIP). The "Digium Asterisk" is one of the commonly used open-source IP-based PBX [182]. It includes all the building blocks needed to create an IP-based PBX, an Interactive Voice Response (IVR) system, a conference bridge and other related communication services. Most common UC deployments with Asterisk use SIP, H.323 and IAX UC protocols. SIP is the most widely used protocol. SIP is an application-layer control (signalling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.

Numerous studies [177, 183–187] discussed on-premises trading communication systems and conventional telephony systems that can converge time-division multiplexing (TDM) and IP-based communication systems, which are more secure and improved performance than the cloud-hosted trading communication system. However, in a trading communication system-based circuit-switched architecture, moving to another technology is impossible as the unique requirements of a trading communication system differ from those generally offered through VoIP [183].

Bakshi et al. [186] demonstrated a software-based trading turret system that could lead to a cloud-based trading communication system. A software-based turret is a step forward in studying the movement of a trading communication system to the Cloud. However, the issues of viability, security, and performance of the trading communication system hosted in the cloud need to be further investigated before real deployments.

This section identified the lack of research focusing on cloud-based UC deployments and assessing cloud-based UC security and performance requirements. It would be beneficial to create a model capable of evaluating any proposed UC system design and implementation before the system is migrated to the cloud. Therefore, the research question is as follows: “Can a cloud-based Trading Communication System achieve security objectives while meeting performance (quality of service) requirements?”.

**Cloud Platform Systems**

OpenStack is an open-source cloud computing project aimed at deployment in all types of cloud environments. Globally, cloud computing experts contribute to this project to make its implementation scalable and straightforward [188].

Numerous studies [147–152] investigate the performance of OpenStack architecture. Recently, the desire of UC the move to a cloud environment has gained more attention [189–191]. Moving to the Cloud requires proper planning and decisions on selecting cloud providers when hosting real-time applications, such as VoIP or video conferencing. The measurement of security and performance capabilities for critical real-time services, such as trading communication systems, is vital.

In previous work [192], we implemented a private cloud environment using the OpenStack platform with high availability and a dynamic resource allocation mechanism. Besides, we implemented UCaaS in the underlying OpenStack platform and experimented with the voice and video call between different parties.

Wen et al. [193] analysed two open-source cloud platforms, OpenStack and OpenNebula, from the perspectives of architecture and security. They need to conduct experimentation to substantiate their proposal. Nasim and Kassler et al. [194] analysed the performance of Openstack in a virtual and physical system. Nasim and Kassler et al. argued that Openstack deployed over dedicated hardware performs better than Openstack running over a virtualised system. The reason is the overhead generated from computational resource usage of the virtual system.

VMware ESXi technology provides pools of servers, storage and networking with dynamically configurable security, availability and management services [195–198]. It is a private or hybrid cloud software solution capable of enabling enterprises to build their multi-tenant private clouds by pooling infrastructure resources into virtual datacenters. Users can access those services through web-based tools. VMware vShield

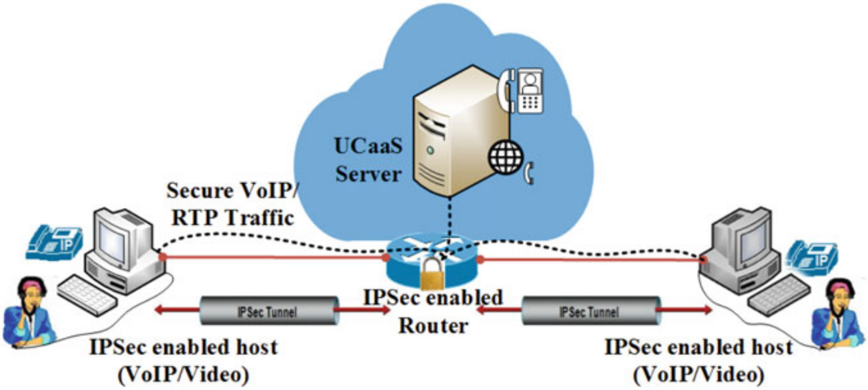


**Fig. 3** Unified communication over IPSec

can also provide comprehensive data and application security, improve visibility and control in a VMware-based cloud [199]. To date, there has been no study to assess vShield's security services in protecting the voice and video communications in a VMware-based Cloud.

Tesfamicael et al. [197] examined the deployment of a cloud system via a VMware suite to emulate cloud-based UC services. They set up an IPSec gateway to support network-level security for UCaaS against possible security exposures. This study aimed to analyse the implementation of UCaaS over IPSec and evaluate the latency of encrypted UC traffic while protecting that traffic. Their test results showed no latency while IPSec is implemented with a G.711 audio codec, as shown in Fig. 3. However, the performance of the G.722 audio codec with an IPSec implementation affected the overall performance of the UC server.

### Summary

Moving Trading Communication Systems to the Cloud may reduce operating costs; however, the operation of a cloud-based Trading Communication System across the Internet can pose many challenges, including performance and security issues. Numerous studies discuss open-source cloud solutions. Nevertheless, a lack of study focuses on deploying UC in the cloud to assess security and real-time performance requirements. More research is needed to address the right balance between security and performance for cloud-based Trading Communication Systems.

## 5 Theme 4: Network Performance and Security Management for Communication System

This section reviews the network performance and security aspects of the VoIP and video conferencing deployments. Whether these services are deployed on-premises or over the Cloud, it is crucial to understand the approach taken in modelling those systems in terms of security and performance [200].

### Video Conferencing

Video conferencing (VC) is a two-way communication that allows participants to interact with each other in real-time in two or more locations over video and audio communication.

Resource allocation for real-time video conferencing has recently been an active research topic [201–203]. Cloud-based real-time video conferencing imposes several constraints on resource management that impact performance. Numerous studies have been conducted regarding resource scheduling in the Cloud.

Cicalo and Tralli [204] proposed cross-layer optimisation of a system for multiple scalable video delivery in wireless networks. Cicalo and Tralli provided a mechanism to minimise the distortion difference among multiple videos. The authors observed differences in efficiency and video quality fairness with different strategies. Cicalo and Tralli study primarily focuses on resource allocation for video transmission in a wireless network environment.

Using a utility maximisation strategy, Chen et al. [199] analysed peer-to-peer video conferencing services. They discovered that the lack of powerful nodes in peer-to-peer systems could not support the execution of high demand video processing tasks. Consequently, they designed a multi-party video conferencing solution called, Celerity, in a peer-to-peer environment. Celerity was designed to optimise the resources of the video conferencing services, which is an advantage over those in other studies [205–208].

Numerous studies [209–211] proposed systems to maximise video quality under bandwidth constraints for peer-to-er multi-party video conferencing. These studies focused on peer-to-peer video conferencing performance analysis, where bandwidth bottlenecks exist only at the edge of the network. The experiment is on the physical communication link rather than video transcoding flexibilities.

Feng et al. [212] maximised the overall throughput of video conferencing sessions to improve the performance of the video services by leveraging more bandwidth using intra-session network coding. Similarly, Mell and Grance [213] presented a number of video scheduling policies for improving the performance and commercial viability of video on demand (VoD) systems.

Generally, video transcoding is a resource-intensive task, and meeting the performance requirements for large video data in the cloud reliably and securely is a challenge. In a typical cloud deployment scenario, a pool of resources for video conferencing is shared. However, modelling is needed to effectively estimate the resource demand for VC services in the cloud, and many studies do not address this. Having a proper modelling mechanism could reduce over-utilised or under-utilised resources and significantly impact the quality and performance of video conferencing.

### Voice-Over-IP (VoIP)

Voice-over-IP is a telecommunication system that allows two or more people to communicate through the Internet. Unlike traditional phone calls, VoIP does not require a public-switched telephone network (PSTN) to make a call.

Numerous studies on open source VoIP have appeared in the last few years. A large portion of the literature on Asterisk reveals its focus on VoIP.

Chirag and Kamaljit [182] presented some considerations concerning an architecture design of an Asterisk server to provide a VoIP system, but the study lacks any assessment of an appropriate test of the system.

Ahmed and Mansor [214] reported practical experiments on the measurement of the central processing unit (CPU) utilisation and performance on an Asterisk VoIP Private Branch Exchange (PBX) performance. They used a traffic generator for the Session Initiation Protocol (SIP), called the SIPp simulator, to generate voice calls to measure the performance of the Asterisk CPU. Their result found that a CPU usage spike incidence increases when concurrent calls are increased. Indeed, the overall performance of Asterisk will generally be affected by the need for large calculations, and thus it is essential to select a computer with a powerful CPU/Floating Point Unit (FPU). However, other experiments have yet to be carried out for other factors, including overall network performance, bandwidth tests for geographically diverse VoIP deployments, and Quality of Service (QoS).

Konstantoulakis and Sloman [215] have successfully implemented a call management policy specific to an Asterisk IP PBX and measured the system's performance when system administrators use the system. Hammoud and Bourget et al. [216] proposed the integration of the Asterisk server with a rule-based engine (InRule) to enable the Asterisk Server to instruct InRule to perform the required analysis. Similarly, Chava and How [217] successfully integrated an Asterisk server with the Cisco Call Manager for interoperability.

Few studies focused on cloud-based VoIP implementation [218, 219]. However, a lack of research focuses on cloud-based UC, particularly VoIP or video conferencing in the open-source cloud platform.

### Quality of Experience (QoE) for UC

QoE can be used to measure user perceived experience for a provided voice service of VoIP system. QoE refers to a user-perceived quality of voice service and is one of the key design influence factors and an indicator for impairments affecting the VoIP quality. QoE basically depends on user satisfaction in terms of usability, accessibility, and integrity of the QoS, which reflects network performance. QoE is not limited to the network's technical performance; non-technical aspects influence user perception and satisfaction with the quality of the service provided by VoIP. Therefore, QoS by itself is incomplete to measure the overall voice quality of the VoIP system without incorporating QoE. QoE and QoS are used to measure the quality of the whole system service.

It is crucial to understand how the network services delivery of VoIP and video conference services are experienced by users, referring to the Quality of Experience (QoE). The difference between QoE and QoS is that the former focuses on a service provisioning paradigm based on how the end-user feels, whereas the latter measures the overall performance of a service from a technology-driven perspective.

Numerous studies have been conducted to investigate QoE for real-time and non-real-time services. Vera et al. [220] and Charonyktakis et al. [221] investigated the QoE assessment approaches for VoIP services by reviewing recent advances related to the QoE for VoIP. Tsolkas et al. [222] provided a guide on how QoE can be standardised and how an actual quality assessment can be conducted mainly for VoIP, online video, video streaming and skype services. Tsolkas et al. [223] studied speech quality estimation by developing a taxonomy of QoE estimation methods.

The ITU-T Standard provides two testing methods, subjective and objective methods of testing voice quality. To study real-time voice and video quality performance, numerous studies [224–228] applied QoE objective and subjective evaluation methods (based on the ITU-T testing standard methods). Some authors [229, 230] focused their study on the impact of the network impairment of a real-time voice system on the voice's overall perceived quality. Some authors [229, 230] discovered that packet loss rate and audio bandwidth are the network characteristics that impact the user's QoE. Laghari and Connelly [231] presented a well-structured detailed taxonomy of QoE focusing on the QoE's business, technical, and human aspects.

Ding et al. [232] proposed a parametric, non-intrusive speech quality assessment algorithm that combines an Internet protocol analysis and the ITU-T E-model. They measured speech quality from a significant VoIP impairment, including packet loss, temporal clipping and noise.

Numerous studies [222, 224, 233, 234] focused on studying the QoE estimation model, Mean Opinion Score (MOS), to assess the quality of media signals, such as video/audio codec, packet loss, mean loss burst size, one–way delay and jitter. Examples of this approach are the Pseudo-Subjective Quality Assessment (PSQA) method [220] and a modular algorithm for user-centric QoE prediction (MLQoE) [221]. The MOS value is usually obtained by interviewing the end-users to evaluate speech quality on a five-point scale (Excellent, Good, Fair, Poor, and Bad) [235]. The MOS model is the most extensively used measurement scale for observations of speech quality, but it should be incorporated with QoS metrics to give an accurate QoE measurement and prediction. In terms of the parametric objective method, most previous studies use the E-model recommended by ITU-T [236, 237].

Hoßfeld and Binzenhöfer [238] performed a QoE assessment of Skype calls over the Universal Mobile Telecommunication System (UMTS) that supports VOIP calls operating in a mobile environment. They analysed the quality of IP-based voice calls using Skype in subjective and objective ways. Subjective methods are usually based on controlled actual experiments with human participants who directly evaluate their experience of a service actively or passively. These methods are empirical in nature. However, in an objective manner, end-user quality is measured or predicted without user intervention and is statistical in nature. Their experiments were based on the performance analysis of measured QoE in a UMTS network.

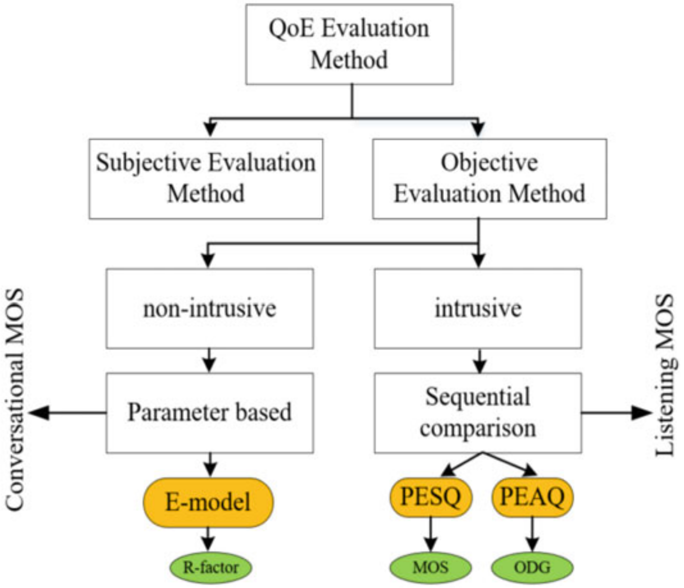


**Fig. 4** QoE evaluation assessment method

Tsolkas et al. [223] and Gómez et al. [239] studied the effect of QoE on the resource efficiency of the system. The authors incorporated a QoE model into the overall network architecture to achieve more resource-efficient operations. Monitoring and controlling the QoE model can determine if investing extra resources will improve the offered services' quality as perceived by the users.

In Fig. 4. an overview of the QoE evaluation assessment method for audio is presented.

## Summary

QoE primarily evaluates the perceived quality of user experience based on a subjective method. Subjective methods are usually based on controlled actual experiments with human participants who directly assess their experience of a service actively or passively. These methods are empirical. However, in an objective approach, quality perceived by end-users is measured or predicted without user intervention and is statistical.

The cloud-based trading system is different from the conventional premises-based VoIP system from a QoE perspective. QoE estimation based on the subjective method is a time-consuming, costly process, inadequate to meet real-time demands, and lacks appropriate reusability. Further, due to the subjective nature of users' ratings, parametric statistical models cannot be applied for QoE measurement and prediction. As such, more work is required in this area to understand the performance requirement of the trading and communication system and model the user QoE accordingly.

## 6 Conclusion

Existing studies focused on peer-to-peer (microgrid) energy trading. Only limited studies can be found in macrogrid energy trading. There is a lack of studies conducted on the feasibility and performance of cloud-based trading communication systems. Therefore, it is essential to provide a model to estimate "quality of service" for a cloud-based trading and communication system and a model to perceive "quality of experience" prior to system deployments.

## References

1. S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, in *Decentralized Business Review*, (2008), p. 21260
2. H. Paik, X. Xu, H.M.N.D. Bandara, S.U. Lee, S.K. Lo, Analysis of data management in blockchain-based systems: from architecture to governance. IEEE Access **7**, 186091–186107 (2019)
3. H. Tang, Y. Shi, P. Dong, Public blockchain evaluation using entropy and TOPSIS. Expert Syst. Appl. **117**, 204–210 (2019)
4. V. Buterin, Ethereum white paper. GitHub Repos. **1**, 22–23 (2013)
5. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in *Proceedings of the Thirteenth EuroSys Conference*, (2018), pp. 1–15
6. Q. Nasir, I.A. Qasse, M. Abu Talib, A.B. Nassif, Performance analysis of hyperledger fabric platforms. Secur. Commun. Netw. 2018 (2018)
7. A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, S. Chatterjee,Performance characterization of hyperledger fabric, in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, (2018), pp. 65–74
8. C. Cachin, Architecture of the hyperledger blockchain fabric, in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, (2016)
9. P. Thakkar, S. Nathan, B. Viswanathan, Performance benchmarking and optimizing hyperledger fabric blockchain platform, in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, (2018), pp. 264–276
10. T.T.A. Dinh, J. Wang, G. Chen, R. Liu, B. Ooi, K. Tan, Blockbench: a framework for analyzing private blockchains, in *Proceedings of the 2017 ACM International Conference on Management of Data*, (2017)
11. C. Gorenflo, S. Lee, L. Golab, S. Keshav, FastFabric: scaling hyperledger fabric to 20,000 transactions per second, in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, (2019), pp. 455–463
12. T. Locher, S. Obermeier, Y.A. Pignolet, When can a distributed ledger replace a trusted third party?, in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, (2018), pp. 1069–1077
13. G.-T. Nguyen, K. Kim, A survey about consensus algorithms used in blockchain. J. Inform. Process. Syst. **14**, 101–128 (2018)
14. C.K. Adiputra, R. Hjort, H. Sato, A proposal of blockchain-based electronic voting system, in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, (2018), pp. 22–27

15. R. Zhang, R. Xue, L. Liu, Security and privacy on blockchain. ACM Comput. Surv. (CSUR) **52**, 1–34 (2019)
16. C. Cachin, M. Vukolić, Blockchain consensus protocols in the wild, (2017). arXiv:1707.01873
17. K. Gai, Y. Wu, L. Zhu, L. Xu, Y. Zhang, Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet Things J. **6**, 7992–8004 (2019)
18. B. Putz, F. Menges, G. Pernul, A secure and auditable logging infrastructure based on a permissioned blockchain. Comput. Secur. **87**, 101602 (2019)
19. T. Hyla, J. Pejaś, eHealth integrity model based on permissioned blockchain. Future Internet **11**, 76 (2019)
20. X. Xiang, M. Wang, W. Fan, A permissioned blockchain-based identity management and user authentication scheme for E-health systems. IEEE Access **8**, 171771–171783 (2020)
21. G.S. Reen, M. Mohandas, S. Venkatesan,Decentralized patient centric e-health record management system using blockchain and ipfs, in *2019 IEEE Conference on Information and Communication Technology*, (2019), pp. 1–7
22. M. Chaieb, M. Koscina, S. Yousfi, P. Lafourcade, R. Robbana, Dabsters: a privacy preserving e-voting protocol for permissioned blockchain, in *International Colloquium on Theoretical Aspects of Computing*, (2019), pp. 292–312
23. N. Faour, Transparent voting platform based on permissioned blockchain, (2018). arXiv:1802.10134
24. K. Garg, P. Saraswat, S. Bisht, S.K. Aggarwal, S.K. Kothuri, S. Gupta, A comparitive analysis on e-voting system using blockchain, in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, (2019), pp. 1–4
25. D. Kirillov, V. Korkhov, V. Petrunin, M. Makarov, I.M. Khamitov, V. Dostov, Implementation of an e-voting scheme using hyperledger fabric permissioned blockchain, in *International Conference on Computational Science and Its Applications*, (2019), pp. 509–521
26. F. Fusco, M.I. Lunesu, F.E. Pani, A. Pinna, Crypto-voting, a blockchain based e-voting system, in *KMIS* (2018), pp. 221–225
27. S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, On the security of permissioned blockchain solutions for IoT applications, in *6th IEEE Conference on Network Softwarization (NetSoft)* (2020), pp. 465–472
28. S. Pal, T. Rabehaja, A. Hill, M. Hitchens, V. Varadharajan, On the integration of blockchain to the internet of things for enabling access right delegation. IEEE Internet Things J. **7**, 2630–2639 (2020)
29. N.M. Kumar, P.K. Mallick, Blockchain technology for security issues and challenges in IoT. Procedia Comput. Sci. **132**, 1815–1823 (2018)
30. M.U. Hassan, M.H. Rehmani, J. Chen, Privacy preservation in blockchain based IoT systems: integration issues, prospects, challenges, and future research directions. Futur. Gener. Comput. Syst. **97**, 512–529 (2019)
31. F. Saleh, Blockchain without waste: proof-of-stake. Rev. Financ. Stud. **34**, 1156–1190 (2020)
32. W. Li, S. Andreina, J.-M. Bohli, G. Karame, Securing proof-of-stake blockchain protocols, in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (Springer, 2017), pp. 297–315
33. N. Houy, It will cost you nothing to'kill'a proof-of-stake crypto-currency (2014). SSRN 2393940
34. R. Angeles, Blockchain-based healthcare-three successful proof-of-concept pilots worth considering. J. Int. Technol. Inform. Manag. **27** (2018)
35. S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain, in *Italian Conference on Cybersecurity* (2018)
36. M. Vukolić, The quest for scalable blockchain fabric: proof-of-work versus BFT replication, in *International Workshop on Open Problems in Network Security* (2016), pp. 112–125
37. Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: architecture, consensus, and future trends. IEEE International Congress on Big Data (BigData Congress) **2017**, 557–564 (2017)

38. B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng et al., Performance analysis and comparison of PoW, PoS and DAG based blockchains. Digit. Commun. Netw. **6**, 480–485 (2020)
39. K. Košt'ál, T. Krupa, M. Gembec, I. Vereš, M. Ries, I. Kotuliak, On transition between PoW and PoS, in *2018 International Symposium ELMAR* (2018), pp. 207–210
40. C. Lepore, M. Ceria, A. Visconti, U.P. Rao, K.A. Shah, L. Zanolini, A survey on blockchain consensus with a performance comparison of PoW, PoS and pure PoS. Mathematics **8**, 1782 (2020)
41. S. Wan, M. Li, G. Liu, C. Wang, Recent advances in consensus protocols for blockchain: a survey. Wireless Netw. **26**, 5579–5593 (2020)
42. A. Li, X. Wei, Z. He, Robust proof of stake: a new consensus protocol for sustainable blockchain systems. Sustainability **12**, 2824 (2020)
43. H. Sukhwani, J.M. Martínez, X. Chang, K.S. Trivedi, A. Rindos, Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric), in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (2017), pp. 253–255
44. L. Feng, H. Zhang, Y. Chen, L. Lou,Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain. Appl. Sci. **8**, 1919 (2018)
45. M. Castro, B. Liskov, Practical Byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. (TOCS) **20**, 398–461 (2002)
46. N. Szabo, *Smart Contracts* (Virtual School, 1994)
47. X. Yang, W.F. Lau, Q. Ye, M.H. Au, J.K. Liu, J. Cheng, Practical escrow protocol for bitcoin. IEEE Trans. Inf. Forensics Secur. **15**, 3023–3034 (2020)
48. Q. Wang, X. Li, Y. Yu, Anonymity for bitcoin from secure escrow address. IEEE Access **6**, 12336–12341 (2017)
49. A. Cohn, T. West, C. Parker, Smart after all: blockchain, smart contracts, parametric insurance, and smart energy grids. Georgetown Law Technol. Rev. **1**, 273–304 (2017)
50. A. Kumari, A. Shukla, R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, ET-DeaL: A P2P smart contract-based secure energy trading scheme for smart grid systems, in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2020), pp. 1051–1056
51. E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, C. Weinhardt, A blockchain-based smart grid: towards sustainable local energy markets. Comput. Sci.-Res. Dev. **33**, 207–214 (2018)
52. F. Schär,Decentralized finance: on blockchain- and smart contract-based financial markets, in *Federal Reserve Bank of St. Louis Review* (vol. 103, 2021), pp. 153–174
53. J. Ellul, G. Pace, Alkylvm: a virtual machine for smart contract blockchain connected internet of things (2018)
54. M.M. Uzair, E. Karim, P. Sultan, S.S. Ahmed, The impact of blockchain technology on the real estate sector using smart contracts, MPRA Paper 88934 (2018)
55. I. Karamitsos, M. Papadaki, N.B. Al Barghuthi, Design of the blockchain smart contract: a use case for real estate. J. Inform. Secur. **9**, 177.
56. C. Fromknecht, D. Velicanu, S. Yakoubov, A decentralized public key infrastructure with identity retention. IACR Cryptol. ePrint Arch. **2014**, 803 (2014)
57. W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, X. Lin, A privacy-preserving thin-client scheme in blockchain-based PKI, in *IEEE Global Communications Conference* (Abu Dhabi, United Arab Emirates, 2018), pp. 1–6
58. S. Misra, S. Goswami, C. Taneja, A. Mukherjee, M.S. Obaidat, A PKI adapted model for secure information dissemination in industrial control and automation 6LoWPANs. IEEE Access **3**, 875–889 (2015)
59. A.D.L.R. Gómez-Arevalillo, P. Papadimitratos, Blockchain-based public key infrastructure for inter-domain secure routing, in *International Workshop on Open Problems in Network Security Rome* (Italy, 2017), pp. 20–38
60. B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, W. Shi, Cecoin: A decentralized PKI mitigating MitM attacks. Fut. Gener. Comput. Syst. (2017)

61. S. Matsumoto, R.M. Reischuk,IKP: turning a PKI around with decentralized automated incentives, in *IEEE Symposium on Security and Privacy* (San Jose, CA, USA, 2017), pp. 410–426
62. L.M. Axon, M. Goldsmith, PB-PKI: a privacy-aware blockchain-based PKI, in *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications Madrid* (Spain, 2017), pp. 311–318.
63. M. Al-Bassam, SCPKI: a smart contract-based PKI and identity system, in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (Abu Dhabi, United Arab Emirates, 2017), pp. 35–40.
64. X. He, J. Lin, K. Li, X. Chen, A novel cryptocurrency wallet management scheme based on decentralized multi-constrained derangement. IEEE Access **7**, 185250–185263 (2019)
65. U. Javaid, B. Sikdar, A lightweight and secure energy trading framework for electric vehicles, in *International Conference on Sustainable Energy and Future Electric Transportation (SEFET)* (2021), pp. 1–6
66. S. Pallickara, G. Fox, NaradaBrokering: a distributed middleware framework and architecture for enabling durable peer-to-peer grids, in *Middleware* (Berlin, 2003), pp. 41–61.
67. R. Oppliger, G. Pernul, C. Strauss, Using attribute certificates to implement role-based authorization and access controls,in *Sicherheit in Informationssystemen (SIS 2000)* (2000), pp. 169–184
68. S.W. Shah, S.S. Kanhere, Recent trends in user authentication—A survey. IEEE Access **7**, 112505–112519 (2019)
69. P. Arias-Cabarcos, C. Krupitzer, C. Becker, A survey on adaptive authentication. ACM Comput. Surv. **52**, Article 80 (2019)
70. B.S. Archana, A. Chandrashekar, A.G. Bangi, B.M. Sanjana, S. Akram, Survey on usable and secure two-factor authentication, in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (2017), pp. 842–846
71. V. Liu, A.D. Tesfamicael, W. Caelli, T. Sahama, Network security metrics and performance for healthcare systems management, in *17th International Conference on E-health Networking, Application & Services (HealthCom)* (2015), pp. 189–194
72. E. Barker, Q. Dang, S. Frankel, K. Scarfone, P. Wouters, Guide to IPsec VPNs (National Institute of Standards and Technology, 2019)
73. S.S. Kolahi, K. Mudaliar, C. Zhang, Z. Gu, Impact of IPSec security on VoIP in different environments, in *Ninth International Conference on Ubiquitous and Future Networks (ICUFN)* (2017), pp. 979–982
74. A.A. Al-khatib, R. Hassan, Impact of IPSec protocol on the performance of network real-time applications: a review. Int. J. Netw. Secur. **20**, 811–819 (2018)
75. A. Sushma, T. Sanguankotchakorn, Implementation of IPsec VPN with SIP softphones using GNS3, in *Proceedings of the 2018 VII International Conference on Network, Communication and Computing* (2018), pp. 152–156
76. F. Bensalah, N. El Kamoun, A. Bahnasse, Evaluation of tunnel layer impact on VOIP performances (IP-MPLS-MPLS VPN-MPLS VPN IPsec). Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **17**, 87 (2017)
77. A. Alharbi, A. Bahnasse, M. Talea, A comparison of VoIP performance evaluation on different environments over VPN multipoint network. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **17**, 123 (2017)
78. N. Thanthry, G. Gopalakrishnan, R. Pendse,Alternate encryption scheme for VoIP traffic, in *43rd Annual 2009 International Carnahan Conference on Security Technology* (2009), pp. 178–183
79. P. Choudhury, K.P. Kumar, S. Nandi, G. Athithan, An empirical approach towards characterization of encrypted and unencrypted VoIP traffic. Multimed. Tools Appl. **79**, 603–631 (2020)
80. M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins et al., Blockchain technology in the energy sector: a systematic review of challenges and opportunities. Renew. Sustain. Energy Rev. **100**, 143–174 (2019)

81. M. Pisa, Reassessing Expectations for blockchain and development. Innov./Blockchain Glob. Dev. **12** (2018)
82. I. El-Sayed, K. Khan, X. Dominguez, P. Arboleya,A real pilot-platform implementation for blockchain-based peer-to-peer energy trading, in *2020 IEEE Power & Energy Society General Meeting (PESGM)* (2020), pp. 1–5
83. M.S. Grewal-Carr V, Blockchain enigma paradox opportunity, in *Deloitte2016* (2016)
84. Blockchain an opportunity for energy producers and consumers? PwC2015
85. N.Z. Aitzhan, D. Svetinovic, Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secur. Comput. **15**, 840–852 (2016)
86. T. Dimitriou, G. Karame, Privacy-friendly planning of energy distribution in smart grids, in *Proceedings of the 2nd Workshop on Smart Energy Grid Security* (Scottsdale, Arizona, USA, 2014)
87. C. Efthymiou, G. Kalogridis,Smart grid privacy via anonymization of smart metering data, in *2010 First IEEE International Conference on Smart Grid Communications* (2010), pp. 238–243
88. F. Li, B. Luo, P. Liu,Secure Information aggregation for smart grids using homomorphic encryption, in *2010 First IEEE International Conference on Smart Grid Communications* (2010), pp. 327–332
89. A. Rial, G. Danezis, Privacy-preserving smart metering, in *Proceedings of the 10th annual ACM Workshop on Privacy in the Electronic Society* (2011), pp. 49–60
90. Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans. Ind. Inf. **14**, 3690–3700 (2018)
91. M.U. Hassan, M.H. Rehmani, J. Chen, DEAL: differentially private auction for blockchain-based microgrids energy trading. IEEE Trans. Serv. Comput. **13**, 263–275 (2020)
92. Q. Wu, M. Zhou, Q. Zhu, Y. Xia, VCG auction-based dynamic pricing for multigranularity service composition. IEEE Trans. Autom. Sci. Eng. **15**, 796–805 (2018)
93. T. Li, W. Zhang, N. Chen, M. Qian, Y. Xu, Blockchain technology based decentralized energy trading for multiple-microgrid systems, in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)* (2019), pp. 631–636
94. A. Paudel, G.H. Beng,A Hierarchical peer-to-peer energy trading in community microgrid distribution systems, in *2018 IEEE Power & Energy Society General Meeting (PESGM)* (2018), pp. 1–5
95. K. Anoh, D. Bajovic, A. Ikpehai, B. Adebisi, D. Vukobratovic, Enabling Peer to Peer Energy Trading in Virtual Microgrids with LP-WAN, in *IEEE EUROCON 2019—18th International Conference on Smart Technologies* (2019), pp. 1–5
96. W. Hua, H. Sun,A blockchain-based peer-to-peer trading scheme coupling energy and carbon markets, in *2019 International Conference on Smart Energy Systems and Technologies (SEST)* (2019), pp. 1–6
97. D. Zhu, B. Yang, Q. Liu, K. Ma, S. Zhu, X. Guan, Joint energy trading and scheduling for multi-energy microgrids with storage, in *2020 39th Chinese Control Conference (CCC)* (2020), pp. 1617–1622
98. H.S.V.S.K. Nunna, A. Sesetti, A.K. Rathore, S. Doolla, Multiagent-based energy trading platform for energy storage systems in distribution systems with interconnected microgrids. IEEE Trans. Ind. Appl. **56**, 3207–3217 (2020)
99. S. Kwak, J. Lee,Implementation of blockchain based P2P energy trading platform, in *2021 International Conference on Information Networking (ICOIN)* (2021), pp. 5–7
100. M.J.A. Baig, M.T. Iqbal, M. Jamil, J. Khan, IoT and blockchain based peer to peer energy trading pilot platform, in *11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (2020), pp. 0402–0406
101. P. Xie, W. Yan, P. Xuan, J. Zhu, Y. Wu, X. Li, et al., Conceptual framework of blockchain-based electricity trading for neighborhood renewable energy, in *2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)* (2018), pp. 1–5

102. S. Johanning, T. Bruckner, Blockchain-based peer-to-peer energy trade: a critical review of disruptive potential, in *16th International Conference on the European Energy Market (EEM)* (2019), pp. 1–8
103. S.J. Pee, E.S. Kang, J.G. Song, J.W. Jang, Blockchain based smart energy trading platform using smart contract, in *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (2019), pp. 322–325
104. H. Wang, J. Huang, Incentivizing energy trading for interconnected microgrids. IEEE Trans. Smart Grid **9**, 2647–2657 (2018)
105. W. Tushar, T.K. Saha, C. Yuen, D. Smith, H.V. Poor, Peer-to-peer trading in electricity networks: an overview. IEEE Trans. Smart Grid **11**, 3185–3200 (2020)
106. J. Bonneau, A. Narayanan, A. Miller, J. Clark, J.A. Kroll, E.W. Felten, Mixcoin: anonymity for bitcoin with accountable mixes, in *Financial Cryptography and Data Security* (Berlin, 2014), pp. 486–504
107. M.C.K. Khalilov, A. Levi, A survey on anonymity and privacy in bitcoin-like digital cash systems. IEEE Commun. Surv. Tutor. **20**, 2543–2585 (2018)
108. L. Valenta, B. Rowan, Blindcoin: blinded, accountable mixes for bitcoin, in *Financial Cryptography and Data Security* (Berlin, 2015), pp. 112–126
109. E. Heilman, L. AlShenibr, F. Baldimtsi, A. Scafuro, S. Goldberg, TumbleBit: an untrusted bitcoin-compatible anonymous payment hub, in *Proceedings 2017 Network and Distributed System Security Symposium* (2017)
110. T. Ruffing, P. Moreno-Sanchez, A. Kate,CoinShuffle: practical decentralized coin mixing for bitcoin, in *Computer Security—ESORICS 2014* (Cham, 2014), pp. 345–364
111. T. Ruffing, P. Moreno-Sanchez, A. Kate, P2P mixing and unlinkable bitcoin transactions, in *NDSS* (2017), pp. 1–15
112. A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: the blockchain model of cryptography and privacy-preserving smart contracts, in *IEEE Symposium on Security and Privacy (SP)* (2016)
113. S. Meiklejohn, R. Mercer,Möbius: trustless tumbling for transaction privacy, in *Proceedings on Privacy Enhancing Technologies* (vol. 2018, 2018), pp. 105–121
114. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G.M. Voelker, et al., A fistful of bitcoins, in *Proceedings of the 2013 Conference on Internet Measurement Conference—IMC '13* (2013)
115. A. Biryukov, D. Feher, G. Vitto, Privacy aspects and subliminal channels in zcash, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom, 2019)
116. Z. Zhang, W. Li, H. Liu, J. Liu, A refined analysis of zcash anonymity. IEEE Access **8**, 31845–31853 (2020)
117. G. Kappos, H. Yousaf, M. Maller, S. Meiklejohn, An empirical analysis of anonymity in zcash, in *USENIX Security Symposium* (Baltimore, MD, USA, 2018), pp. 463–477
118. A. Biryukov, S. Tikhomirov, Security and privacy of mobile wallet users in bitcoin, dash, monero, and zcash. Pervasive Mob. Comput. **59**, 101030 (2019)
119. S.-F. Sun, M.H. Au, J.K. Liu, T.H. Yuen, RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero, in *Computer Security—ESORICS 2017* (Cham, 2017), pp. 456–474
120. B. Bünz, S. Agrawal, M. Zamani, D. Boneh, Zether: towards privacy in a smart contract world, in *International Conference on Financial Cryptography and Data Security* (Cham, 2020), pp. 423–443
121. A. Rondelet, M. Zajac, Zeth: on integrating zerocash on ethereum (2019). arXiv:1904.00905
122. H. Zhao, X. Bai, S. Zheng, L. Wang, RZcoin: ethereum-based decentralized payment with optional privacy service. Entropy **22**, 712 (2020)
123. S. Somin, G. Gordon, Y. Altshuler, Network analysis of erc20 tokens trading on ethereum blockchain, in *Proceedings of the Ninth International Conference on Complex Systems* (Cambridge, MA, USA, 2018), pp. 439–450

124. M. Spagnuolo, F. Maggi, S. Zanero, *BitIodine: extracting intelligence from the bitcoin network*, in *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2014), pp. 457–468
125. D. Ron, A. Shamir, Quantitative analysis of the full bitcoin transaction graph, in *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2013), pp. 6–24
126. E. Androulaki, G.O. Karame, M. Roeschlin, T. Scherer, S. Capkun, Evaluating user privacy in bitcoin, in *Financial Cryptography and Data Security* (Berlin, Heidelberg, 2013), pp. 34–51
127. F. Reid, M. Harrigan, An analysis of anonymity in the bitcoin system, in *Security and Privacy in Social Networks* (2013), pp. 197–223
128. I. Miers, C. Garman, M. Green, A.D. Rubin, Zerocoin: anonymous distributed E-cash from bitcoin, in *IEEE Symposium on Security and Privacy* (2013), pp. 397–411.
129. E.B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, et al., Zerocash: decentralized anonymous payments from bitcoin, in *IEEE Symposium on Security and Privacy* (2014), pp. 459–474
130. M. Mihaylov, S. Jurado, K. Van Moffaert, N. Avellana, A. Nowe, NRG-X-change a novel mechanism for trading of renewable energy in smart grids, in *SMARTGREENS 2014—in International Conference on Smart Grids and Green IT Systems* (2014), pp. 101–106
131. M. Mihaylov, S. Jurado, N. Avellana, K.V. Moffaert, I.M.d. Abril, A. Nowé, NRGcoin: virtual currency for trading of renewable energy in smart grids, in *11th International Conference on the European Energy Market (EEM14)* (2014), pp. 1–6
132. A. Laszka, A. Dubey, M. Walker, D. Schmidt, Providing privacy, safety, and security in IoT-based transactive energy systems using distributed ledgers, in *Proceedings of the Seventh International Conference on the Internet of Things* (Linz, Austria, 2017), pp. 1–8
133. N. Kaaniche, M. Laurent, A blockchain-based data usage auditing architecture with enhanced privacy and availability, in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)* (2017), pp. 1–5
134. J. Schmitz, G. Leoni, Accounting and auditing at the time of blockchain technology: a research agenda. Aust. Account. Rev. **29**, 331–342 (2019)
135. X. Zhu, D. Wang, Application of blockchain in document certification, asset trading and payment reconciliation. J. Phys.: Conf. Ser. **1187**, 052080 (2019)
136. C. Ingle, A. Samudre, P. Bhavsar, P.S. Vidap, Audit and compliance in service management using blockchain, in *2019 IEEE 16th India Council International Conference (INDICON)* (2019), pp. 1–4
137. A.M. Rozario, M. Vasarhelyi, Auditing with smart contracts. Int. J. Digital Account. Res. **18**, 1–27 (2018)
138. E. Bonsón, M. Bednarova, Blockchain and its implications for accounting and auditing. Meditari Account. Res. (2019)
139. V.L. Lemieux, D. Hofman, D. Batista, A. Joo, Blockchain technology for recordkeeping. ARMA Int. Educ. Found. (2019)
140. P.W. Abreu, M. Aparicio, C.J. Costa, Blockchain technology in the auditing environment, in *13th Iberian Conference on Information Systems and Technologies (CISTI)* (2018), pp. 1–6
141. Y. Rechtman, Blockchain: the making of a simple, secure recording concept: certified public accountant. CPA J. **87**, 15–17 (2017)
142. W. Zhang, Y. Yuan, Y. Hu, K. Nandakumar, A. Chopra, A. Caro, Blockchain-based distributed compliance in multinational corporations' cross-border intercompany transactions, in *Future of Information and Communication Conference* (Cham, 2018), pp. 304–320
143. X. Chu, T. Jiang, X. Li, X. Ding, Bye audit! a novel blockchain-based automated data processing scheme for bank audit confirmation, in *CCF China Blockchain Conference* (2020), pp. 68–82
144. S. Wohlgemuth, K. Umezawa, Y. Mishina, K. Takaragi, Competitive compliance with blockchain, in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (2019), pp. 967–972
145. L. Ismail, H. Materwala, S. Zeadally, Lightweight blockchain for healthcare. IEEE Access **7**, 149935–149951 (2019)

146. A. Rozario, C. Thomas, Reengineering the audit with blockchain and smart contracts. J. Emerg. Technol. Account **16** (2019)
147. J.R. Raphael, Rethinking the audit: innovation is transforming how audits are conducted—And even what it means to be an auditor. J. Account. **223**, 28 (2017)
148. J.A. Jaoude, R.G. Saade, Blockchain applications—Usage in different domains. IEEE Access **7**, 45360–45381 (2019)
149. J. Dai, M. Vasarhelyi, Toward blockchain-based accounting and assurance. J. Inform. Syst. **31** (2017)
150. H.-L. Nguyen, C.-L. Ignat, O. Perrin,Trusternity: auditing transparent log server with blockchain, in *Companion Proceedings of the Web Conference 2018* (2018), pp. 79–80
151. T. Antipova, Using blockchain technology for government auditing, in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)* (2018), pp. 1–6
152. J. Dai, N. He, H. Yu, Utilizing blockchain and smart contracts to enable audit 4.0: from the perspective of accountability audit of air pollution control in China. J. Emerg. Technol. Account. **16** (2019)
153. S.S. Smith, Blockchain augmented audit—Benefits and challenges for accounting professionals. J. Theor. Account. Res. **14**, 117–137 (2018)
154. A. Sutton, R. Samavi, Blockchain enabled privacy audit logs, in *International Semantic Web Conference* (2017), pp. 645–660
155. B. Nathalie, G. Marion, M. Jean-Henry, S. Arbër, The potential impact of blockchain technology on audit practice. J. Strateg. Innov. Sustain. **14**, 35–59 (2019)
156. S. Suzuki, J. Murai, Blockchain as an audit-able communication channel, in *IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)* (2017), pp. 516–522
157. A. Kwilinski, Implementation of blockchain technology in accounting sphere. Acad. Account. Financ. Stud. J. **23**, 1–6 (2019)
158. Y. Zhang, F. Xiong, Y. Xie, X. Fan, H. Gu, The impact of artificial intelligence and blockchain on the accounting profession. IEEE Access **8**, 110461–110477 (2020)
159. J. Xue, C. Xu, Y. Zhang, L. Bai, DStore: a distributed cloud storage system based on smart contracts and blockchain, in *International Conference on Algorithms and Architectures for Parallel Processing* (2018), pp. 385–401
160. Blockchain Technology: A Game-changer in accounting?, Deloitte. https://www2.deloitte.com/content/dam/Deloitte/de/Documents/Innovation/Blockchain_A%20game-changer%20in%20accounting.pdf. Accessed 10 May 2021
161. N. Rückeshäuser, Do we really want blockchain-based accounting? Decentralized consensus as enabler of management override of internal controls, in *International Tagung Wirtschafts informatik* (2017), pp. 16–30
162. S. Kozlowski, An audit ecosystem to support blockchain-based accounting and assurance, in *Continuous Auditing* (Emerald Publishing Limited, 2018), pp. 299–313
163. R. Hugh, A. Brian, R. Megan, Q&A. Is internal audit ready for blockchain? Technol. Innov. Manag. Rev. **7** (2017)
164. Y. Wang, A. Kogan, Designing confidentiality-preserving blockchain-based transaction processing systems. Int. J. Account. Inf. Syst. **30**, 1–18 (2018)
165. J. Kokina, R. Mancha, D. Pachamanova, Blockchain: emergent industry adoption and implications for accounting. J. Emerg. Technol. Account. **14**, 91–100 (2017)
166. Rubix—Deloitte Enterprise Blockchain Government Solutions?, Deloitte. https://bitcoinexchangeguide.com/rubix/. Accessed 5 May 2021
167. KPMG and Microsoft announce new “blockchain nodes”, KPMG. https://home.kpmg/sg/en/home/media/press-releases/2017/02/kpmgand-microsoft-announce-new-blockchain-nodes.html. Accessed 5 May 2021
168. C.A.P.A.K. James, R. Kozloski, Weldemariam, Blockchain-enhanced mobile telecommunication device, USA Patent (2018)
169. A.D. Bolton, L. Goosen, E. Kritzinger, Unified communication technologies at a global automotive organization, in *Encyclopedia of Organizational Knowledge, Administration, and Technology* (IGI Global, 2021), pp. 2592–2608

170. A. Teckchandani, Slack: a unified communications platform to improve team collaboration (Academy of Management Briarcliff Manor, NY, 2018)
171. J. Baraković Husić, S. Baraković, E. Cero, N. Slamnik, M. Oćuz, A. Dedović, et al., Quality of experience for unified communications: a survey. Int. J. Netw. Manag. **30**, e2083 (2020)
172. D. Evans, An introduction to unified communications: challenges and opportunities. ASLIB Proc. **56**, 308–314 (2004)
173. A. Tesfamicael, V. Liu, B. Caelli, Design, implementation and evaluation of unified communications on-premises and over the cloud. Int. J. Web Sci. Eng. Smart Dev. **2**, 1–18 (2015)
174. J. Palonka, T. Porębska-Miąc, Cloud computing and mobility as the main trends in unified communications. Studia Ekonomiczne **188**, 119–134 (2014)
175. D. Dziembek, T. Turek, Characteristics and application of unified communications as a service (UCaaS) in enterprises, Informatyka Ekonomiczna, 47–65 (2018)
176. L.P.K. Rehor, A. Hutton, R. Jain, Use cases and requirements for SIP-based media recording (SIPREC) (2011)
177. J. Penton, A. Terzoli, Asterisk: a converged tdm and packet-based communications system, in *Proceedings of SATNAC 2003-Next Generation Networks* (2003)
178. R. Rizky, Z. Hakim, Analysis and design of voip server (voice internet protocol) using asterisk in statistics and statistical informatics communication of Banten province using Ppdioo method. J. Phys.: Conf. Ser. 012160 (2019)
179. S. Khan, N. Sadiq, Design and configuration of VoIP based PBX using asterisk server and OPNET platform, in *International Electrical Engineering Congress (iEECON)* (2017), pp. 1–4
180. D. Pal, T. Triyason, V. Vanijja, Asterisk server performance under stress test, in *IEEE 17th International Conference on Communication Technology (ICCT)* (2017), pp. 1967–1971
181. P. Nuño, C. Suárez, E. Suárez, F.G. Bulnes, F.J. delaCalle, J.C. Granda, A diagnosis and hardening platform for an asterisk VoIP PBX. Secur. Commun. Netw. **2020** (2020)
182. K.G. Chirag, I.L. Kamaljit, Implement VoIP based IP telephony with open source asterisk architecture. Int. J. Interdiscip. Telecommun. Netw. (IJITN) **2**, 1–11 (2010)
183. A. Mallard, From the telephone to the economic exchange: how small businesses use the telephone in their market relations. Environ. Plan. D: Soc. Space **22**, 117–134 (2004)
184. F. Muniesa, Trading-room telephones and the identification of counterparts, in *Living in a Material World* (2008), pp. 291–315
185. A.D. Tesfamicael, V. Liu, W. Caelli, J. Zureo, Implementation and evaluation of open source unified communications for SMBs, in *International Conference on Computational Intelligence and Communication Networks* (2014), pp. 1243–1248
186. A. Bakshi, R. Jain, A.G. Klaiber, K.N. Udall, R.K. Vankayala, Software based trading turret. Google Patents (2012)
187. S.J. Minutillo, A. Bakshi, R. Jain, Converged desktop between a PC and a trading turret. Google Patents (2013)
188. What is OpenStack?, OpenStack. https://docs.openstack.org/icehouse/. Accessed 12 Feb 2021
189. D.S. Linthicum, Cloud computing changes data integration forever: what's needed right now. IEEE Cloud Comput. **4**, 50–53 (2017)
190. H. Bangui, S. Rakrak, S. Raghay, B. Buhnova, Moving to the edge-cloud-of-things: recent advances and future research directions. Electronics **7**, 309 (2018)
191. O. Zimmermann, Architectural refactoring for the cloud: a decision-centric view on cloud migration. Computing **99**, 129–145 (2017)
192. A.D. Tesfamicael, V. Liu, W. Caelli, Design and implementation of unified communications as a service based on the open stack cloud environment, in *IEEE International Conference on Computational Intelligence & Communication Technology* (2015), pp. 117–122
193. X. Wen, G. Gu, Q. Li, Y. Gao, X. Zhang, Comparison of open-source cloud management platforms: openstack and OpenNebula, in *9th International Conference on Fuzzy Systems and Knowledge Discovery* (2012), pp. 2457–2461

194. R. Nasim, A.J. Kassler, Deploying openstack: virtual infrastructure or dedicated hardware, in *IEEE 38th International Computer Software and Applications Conference Workshops* (2014), pp. 84–89
195. K. Sharma, An alleviated model for private cloud deployment using VMware, in *International Conference on Information, Communication, Instrumentation and Control (ICICIC)* (2017), pp. 1–3
196. D. Kargatzis, S. Sotiriadis, E.G. Petrakis, Virtual machine migration in heterogeneous clouds: from openstack to VMWare, in *IEEE 38th Sarnoff Symposium* (2017), pp. 1–6
197. A.D. Tesfamicael, V. Liu, W. Caelli, performance analysis of secure unified communications in the VMware-based cloud, in *International Conference on Computational Intelligence and Communication Networks (CICN)* (2015), pp. 1135–1140
198. J.P. Walters, A.J. Younge, D.I. Kang, K.T. Yao, M. Kang, S.P. Crago, et al., GPU passthrough performance: a comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL applications, in *IEEE 7th International Conference on Cloud Computing* (2014), pp. 636–643
199. R.D. Zota, I.A. Petre, An overview of the most important reference architectures for cloud computing. Inform. Econ. **18**, 26–39 (2014)
200. A.D. Tesfamicael, V. Liu, E. Foo, W. Caelli, Modeling for performance and security balanced trading communication systems in the cloud, in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)* (2017), pp. 1–7
201. Z. Liao, L. Zhang, Scheduling dynamic multicast requests in advance reservation environment for enterprise video conferencing systems. IEEE access **8**, 76913–76928 (2020)
202. R.G. Clegg, R. Landa, D. Griffin, M. Rio, P. Hughes, I. Kegel et al., Faces in the clouds: long-duration, multi-user, cloud-assisted video conferencing. IEEE Trans. Cloud Comput. **7**, 756–769 (2019)
203. A.A. Khalek, C. Caramanis, R.W. Heath, Delay-constrained video transmission: quality-driven resource allocation and scheduling. IEEE J. Select. Top. Signal Process. **9**, 60–75 (2015)
204. S. Cicalo, V. Tralli, Distortion-fair cross-layer resource allocation for scalable video transmission in OFDMA wireless networks. IEEE Trans. Multimed. **16**, 848–863 (2014)
205. J. Li, P. A. Chou, C. Zhang, Mutualcast: an efficient mechanism for content distribution in a peer-to-peer (P2P) network, Microsoft Res. MSR-TR-2004 **100** (2004)
206. M. Chen, M. Ponec, S. Sengupta, J. Li, P.A. Chou, Utility maximization in peer-to-peer systems with applications to video conferencing. IEEE/ACM Trans. Netw. **20**, 1681–1694 (2012)
207. X. Chen, M. Chen, B. Li, Y. Zhao, Y. Wu, J. Li, Celerity: a low-delay multi-party conferencing solution, in *Proceedings of the 19th ACM international conference on Multimedia* (2011), pp. 493–502
208. M. Ponec, S. Sengupta, M. Chen, J. Li, P.A. Chou, Multi-rate peer-to-peer video conferencing: a distributed approach using scalable coding, in *IEEE International Conference on Multimedia and Expo* (2009), pp. 1406–1413
209. M. Ponec, S. Sengupta, M. Chen, J. Li, P.A. Chou, Optimizing multi-rate peer-to-peer video conferencing applications. IEEE Trans. Multimed. **13**, 856–868 (2011)
210. W. Zhu, C. Luo, J. Wang, S. Li, Multimedia cloud computing. IEEE Signal Process. Mag. **28**, 59–69 (2011)
211. D. Ghose, H.J. Kim, Scheduling video streams in video-on-demand systems: a survey. Multimed. Tools Appl. **11**, 167–195 (2000)
212. Y. Feng, B. Li, B. Li, Airlift: video conferencing as a cloud service using inter-datacenter networks, in *20th IEEE International Conference on Network Protocols (ICNP)* (2012), pp. 1–11
213. P. Mell, T. Grance, The NIST definition of cloud computing (2011)
214. M. Ahmed, A.M. Mansor,CPU dimensioning on performance of asterisk VoIP PBX, in *11th Communications and Networking Simulation Symposium* (Ottawa, Canada, 2008), pp. 139–146

215. G. Konstantoulakis, M. Sloman, Call management policy specification for the asterisk telephone private branch exchange, in *Eighth IEEE International Worskshop on Policies for Distributed Systems and Networks* (2007), pp. 251–255
216. A. Hammoud, D. Bourget, Integrating asterisk with in rule to detect suspicious calls, in *Sixth Advanced International Conference on Telecommunications* (2010), pp. 153–160
217. K.S. Chava, J. Ilow, Integration of open source and enterprise IP PBXs, in *3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities* (2007), pp. 1–6
218. A. Tchernykh, J.M. Cortés-Mendoza, I. Bychkov, A. Feoktistov, L. Didelot, P. Bouvry et al., Configurable cost-quality optimization of cloud-based VoIP. J. Parall. Distrib. Comput. **133**, 319–336 (2019)
219. M.M. Abualhaj, M.M. Al-Tahrawi, S.N. Al-Khatib,Performance evaluation of VoIP systems in cloud computing. J. Eng. Sci. Technol. **14**, 1398–1405 (2019)
220. D.D. Vera, P. Rodríguez-Bocca, G. Rubino, Automatic quality of experience measuring on video delivering networks. SIGMETRICS Perform. Eval. Rev. **36**, 79–82 (2008)
221. P. Charonyktakis, M. Plakia, I. Tsamardinos, M. Papadopouli, On user-centric modular QoE prediction for VoIP based on machine-learning algorithms. IEEE Trans. Mob. Comput. **15**, 1443–1456 (2015)
222. D. Tsolkas, E. Liotou, N. Passas, L. Merakos, A survey on parametric QoE estimation for popular services. J. Netw. Comput. Appl. **77**, 1–17 (2017)
223. D. Tsolkas, E. Liotou, N. Passas, L. Merakos, The need for QoE-driven interference management in femtocell-overlaid cellular networks, in International Conference on Mobile and Ubiquitous Systems: computing, Networking, and Services (2013), pp. 588–601
224. T. Daengsi, P. Wuttidittachotti, QoE modeling for voice over IP: simplified e-model enhancement utilizing the subjective MOS prediction model: a case of G. 729 and Thai users. J. Netw. Syst. Manag. **27**, 837–859 (2019)
225. C. Sloan, N. Harte, D. Kelly, A.C. Kokaram, A. Hines, Objective assessment of perceptual audio quality using ViSQOLAudio. IEEE Trans. Broadcast. **63**, 693–705 (2017)
226. J. van der Hooft, M.T. Vega, C. Timmerer, A.C. Begen, F. De Turck, R. Schatz, Objective and subjective QoE evaluation for adaptive point cloud streaming, in *Twelfth International Conference on Quality of Multimedia Experience (QoMEX)* (2020), pp. 1–6
227. T. Abar, A.B. Letaifa, S. El Asmi, Objective and subjective measurement QoE in SDN networks, in *13th International Wireless Communications and Mobile Computing Conference (IWCMC)* (2017), pp. 1401–1406
228. A.H. Mohseni, A. Jahangir, S. Hosseini, Toward a comprehensive subjective evaluation of VoIP users' quality of experience (QoE): a case study on Persian language. Multimed. Tools Appl. 1–20 (2021)
229. A.D. Tesfamicael, V. Liu, E. Foo, B. Caelli, QoE estimation model for a secure real-time voice communication system in the cloud, in *Proceedings of the Australasian Computer Science Week Multiconference* (2019), pp. 1–9
230. Z. Hu, H. Yan, T. Yan, H. Geng, G. Liu, Evaluating QoE in VoIP networks with QoS mapping and machine learning algorithms. Neurocomputing **386**, 63–83 (2020)
231. K.U.R. Laghari, K. Connelly, Toward total quality of experience: A QoE model in a communication ecosystem. IEEE Commun. Mag. **50**, 58–65 (2012)
232. L. Ding, Z. Lin, A. Radwan, M.S. El-Hennawey, R.A. Goubran,Non-intrusive single-ended speech quality assessment in VoIP. Speech Commun. **49**, 477–489 (2007)
233. T. Hoßfeld, P.E. Heegaard, M. Varela, L. Skorin-Kapov, Confidence interval estimators for MOS values (2018). arXiv:1806.01126
234. J. Hosek, P. Vajsar, L. Nagy, M. Ries, O. Galinina, S. Andreev, et al., Predicting user QoE satisfaction in current mobile networks, in *IEEE International Conference on Communications (ICC)* (2014), pp. 1088–1093
235. Methods for subjective determination of transmission quality, ITU-T. https://www.itu.int/rec/T-REC-P.800-199608-I. Accessed 15 July 2021

236. G. 107: the E-model: a computational model for use in transmission planning. https://www.itu.int/rec/T-REC-G.107. Accessed 15 July 2021
237. G. 108: application of the E-model: a planning guide. https://www.itu.int/rec/T-REC-G.108. Accessed 16 July 2021
238. T. Hoßfeld, A. Binzenhöfer, Analysis of skype VoIP traffic in UMTS: end-to-end QoS and QoE measurements. Comput. Netw. **52**, 650–666 (2008)
239. G. Gómez, J. Lorca, R. García, Q. Pérez, Towards a QoE-driven resource control in LTE and LTE-A networks. J. Comput. Netw. Commun. **2013** (2013)

# DDoS Threats and Solutions for 5G-Enabled IoT Networks

**Daniel Onoja, Michael Hitchens, and Rajan Shankaran**

**Abstract** In recent years, the need for seamless connectivity has increased across various network platforms like IoT, with demands coming from industries, homes, mobile, transportation and office networks. The 5th generation (5G) network is being deployed to meet such demand for high-speed seamless network device connections. 5G is a high-speed network technology with a seamless connection of different network devices in an internet of things (IoT) network area. However, the advantages of 5G also contribute to the security challenges. The seamless connectivity 5G provides could be a security threat allowing attacks such as distributed denial of service (DDoS) because attackers might have easy access to the network infrastructure and higher bandwidth to enhance the effects of the attack. We look at DDoS attacks and the classification of DDoS. We discuss some general approaches proposed to mitigate DDoS threats. This paper covers approaches using SDN in 5G enabled IoT network platforms.

**Keywords** DDoS · 5G · IoT · SDN · Bandwidth · Network resources

## 1 Introduction

The need for seamless connection of different devices has experienced a tremendous increase over the years. Connections between devices across several platforms like industry, home, mobile, office, and transport networks have become a thing of necessity. With this comes the demand for efficient, high-speed connectivity. The new generation network being deployed to meet such demand is the fifth generation (5G) infrastructure. It is a very exciting period for the telecommunications industry with

D. Onoja (✉) · M. Hitchens · R. Shankaran
School of Computer Science, Macquarie University, Sydney, NSW 2109, Australia
e-mail: daniel.onoja@hdr.mq.edu.au

M. Hitchens
e-mail: michael.hitchens@mq.edu.au

R. Shankaran
e-mail: rajan.shankaran@mq.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_5

the deployment of 5G and the increased need for high-speed connectivity and accessibility for end-user consumers. The development and deployment of 5G networks are also increasingly becoming the source of advanced technology in the Internet of Things (IoT) applications and other network platforms [50], but with all such advanced networking and telecommunications technology comes potential security risks.

5G is a high-speed network technology that facilitates seamless connectivity between different devices over a large geographical area. This new technology brings immense benefits, but at the same time, it is vulnerable to novel security challenges and threats. For instance, the seamless connectivity feature of 5G is susceptible to Distributed Denial of Service (DDoS) attacks as malicious actors can employ the enhanced connection and bandwidth attributes of the network infrastructure. Security solutions are required to safeguard 5G networks. The 5G network architecture consists of two main parts, which are the new radio (NR) network and the 5G core network (5GC) [51]. The NR has a larger geographical range which can be used for connection between devices and the network [24]. Both network types have been enhanced considerably in comparison to the previous generations (4G, 3G) technology.

Software Defined Network (SDN) is a key enabling technology in the 5G framework as it offers a central control plane that is used in threat detection and attack mitigation by monitoring the entire network. SDN creates a logically centralized control structure of the network by separating the control panel from the forwarding plane of the network [36]. The flexibility and adaptability of SDN make it an ideal candidate for implementing 5G architectures. It is important to ensure that the SDN infrastructure can respond to security threats such as DDoS.

The Internet of Things (IoT) is an emerging paradigm that connects a large number of smart devices with computation and network capabilities through the internet [7]. IoT applications provide seamless integration of information and communication technologies between the cyber-world and the physical environment (classified as smart cities). IoT services support many large distributed systems. A connected transportation system, for instance, could have many devices for controlling traffic signals, sensing, and communicating with vehicles deployed throughout a city. A large car production company will need to ensure the security and safety of millions of cars on the road. A smart grid could consist of networked sub-systems for metering, data collection, data aggregation, and energy distribution. An oil and gas company may need to interconnect hundreds of remote sites such as oil rigs, refineries, exploration sites and pipelines [14]. These devices enhance hospitals, industries, cities, and our homes. These very advantages and critical services and applications mean that security vulnerability in IoT systems is an issue that needs to be considered as such threats could lead to consumer dissatisfaction, security bridge, data or privacy violation, physical attacks, home invasions and malicious attacks such as distributed denial of service (DDoS) attacks.

A Denial of Service Attack (DoS) is when an attacker targets a network with the intention of disrupting the network or network services by consuming the network resources to prevent legitimate users from accessing such resources [54]. In a DoS attack, an attacker uses a device to attack a specific server or network [21]. However, in a DDoS attack, the attacker uses multiple devices/sources to target a network system or resources to disrupt the network or services by consuming the network resources to prevent legitimate users from accessing such resources [54]. DDoS attacks are among the most common cyber security threats to network infrastructure or the internet [19]. A DDoS attack can significantly disrupt the service or performance of network systems or servers by overwhelming them with multiple requests. In DDoS, A DDoS attack is more sophisticated and severe than a DoS attack as the former will consume the network resources faster than the latter and be more difficult to counteract.

In most DDoS attacks, the attacker uses malicious software to launch an attack on a target machine. An attacker will develop a malware program and distribute this over the internet, often with the aid of a website or email attachment. When a vulnerable device is used to visit these websites or open these infected email attachments, the malware will be installed without the knowledge or consent of the user. The device has now become part of the attacker's army of infected computers. This army is called a botnet that could range from 100 to 1000 s of devices worldwide. These botnets wait for instructions to come from the attacker or master device. Once the attacker sends an instruction to attack a targeted server, all the devices in the botnet will attack without the knowledge of the owners of the subverted devices.

The rest of this paper is organized as follows. Section 2 provides background information on DDoS attacks and the proposed mechanism against DDoS attacks. In Sect. 3, we discuss security solutions proposed in 5G/IoT networks against DDoS attacks and conclude the paper in Sect. 4.

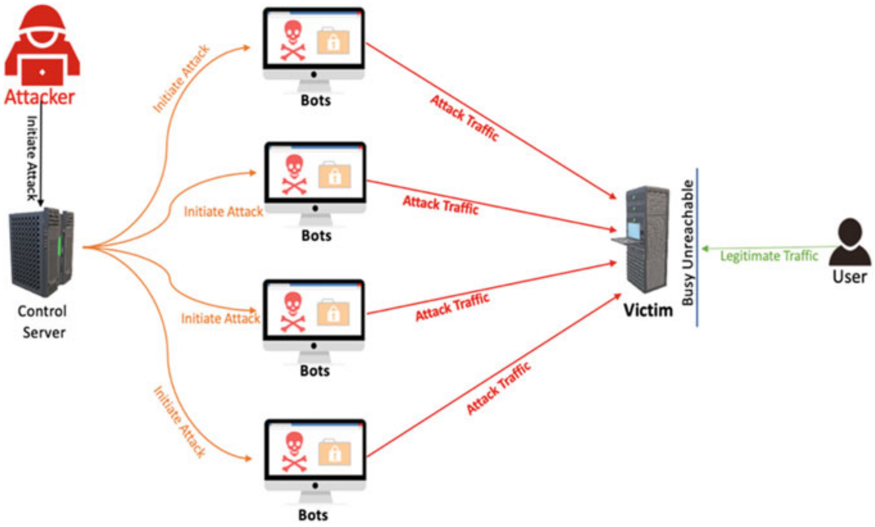


**Fig. 1** DDoS re-illustration

## 2 DDoS Attack Mitigation Categories

Defense mechanisms against a DDoS attack can be separated into three categories. These categories are: attack prevention schemes, attack detection approaches and attack mitigation techniques. However, before we expound on these categories, we will need to understand the type of DDoS attacks by analyzing DDoS classifications.

### *2.1 Classification of DDoS Attack*

To understand what is happening to the network system, it is necessary to analyze the classification of DDoS attacks. There are two major categories of DDoS attacks: bandwidth depletion and resource depletion [10].

**Bandwidth Depletion Attacks**. These are volumetric attacks designed to generate massive unwanted traffic to flood the victim network, making it impossible for legitimate traffic to reach the network system. There are two main classes of bandwidth depletion attack: Flooding attack and amplified attack. The most common example of a DDoS attack is a flooding attack. In this type of attack, the attacker sends large numbers of packets or traffic to the network, continues to reduce the network's performance and denies or limits access to the network [37]. The effect of this attack occurs when the network is overloaded with these large traffic volumes that the network

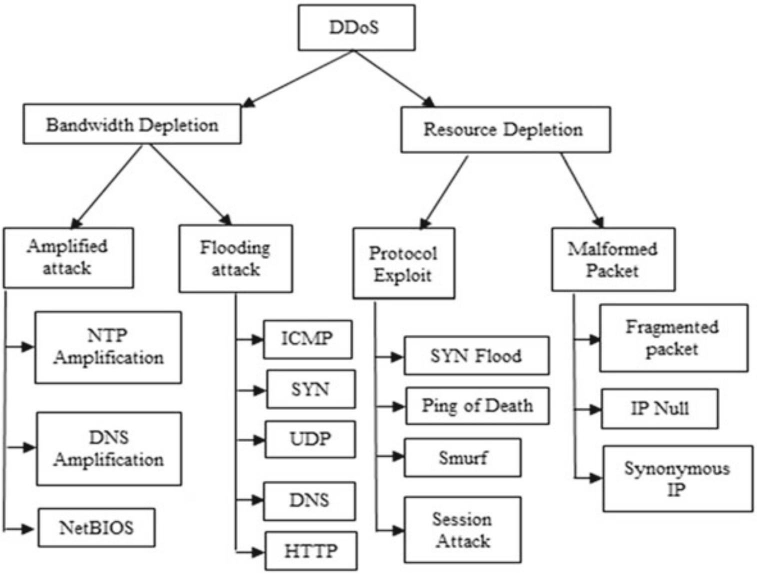


**Fig. 2** Classification of DDoS

resources are overused and unavailable for legitimate users. Flooding attacks can be divided into subclasses; ICMP flood, SYN flood, UDP flood, DNS flood, HTTP flood and amplified attack can be divided into subclasses; NTP amplification, DNS amplification and NetBIOS.

**Resource Depletion Attack**. An attack is designed to target a network system's resources by exhausting the network resources. Resource depletion attacks can be divided into two main classes: protocol exploit and malformed packet. Protocol exploit can be divided into subclasses; Ping of Death, SYN flood, smurf, session attack and malformed packets can be divided into subclasses; fragmented packet, IP Null, and synonymous IP.

In recent years, DDoS attacks have increased in frequency, sophistication, and severity [42]. The severity of the DDoS attacks on large network systems has increased the need for better mitigation mechanisms against such attacks, a significant part of any security planning.

The rest of this paper is organized as follows: Section 2 provides a general literature review and background information on the mechanism proposed to combat DDoS attacks. Section 3 discusses specific approaches proposed against DDoS attacks in 5G/IoT and concludes the paper in Sect. 4.

## *2.2 Attack Prevention*

Dutch philosopher Desiderius Erasmus often credited the saying "prevention is better than cure" is often credited to Dutch philosopher Desiderius Erasmus. Loosely, it means that it is better to stop something bad from happening than to deal with it after it has happened [48]. Preventing an attack could be more cost-effective than the damages an attack can cause to a network system. Attack prevention schemes are designed to prevent attacks from reaching the target system. Prevention schemes can be deployed anywhere in the network. However, they are mostly deployed at the network edge (routers, switches, and end-user devices). Many attack prevention schemes are based on filters, but other approaches include resource disabling and involve the use of honeypots.

**Filters**. These prevention schemes are mostly installed at the network edge, including packet filtering mechanisms configured to filter the traffic only to allow identified and valid traffic to pass through the routers. As much as the mechanism could be efficient in filtering one particular traffic, there is still the challenge of configuring a specific filtering rule that will apply to all traffics flow types to accurately differentiate a spoofed one from a legitimate one without any false positives. In a DDoS attack, most attackers tend to spoof legitimate traffic flow to conceal the real identity of the source address of the attack traffic. Attack prevention schemes could be configured with the assumption that some malicious traffics are spoofed. Several filtering mechanisms have been proposed over the years to safeguard against such DDoS attacks.

*Ingress/Egress Filtering*. Ferguson and Senie [41] is the most basic prevention filtering mechanism against DoS attacks. It is a filtering mechanism used to filter the traffic or packets coming into the network *(Ingress)* or leaving the network *(Egress)*. In both *Ingress and Egress* approaches, the filters are configured with a specific system or network requirements. Any traffic or packets that do not meet these requirements will be blocked or dropped at the edge of the network. *Routers use ingress filtering mechanisms* for filtering packet traffic entering the local network or target, while the *Egress filtering* mechanism is used to filter packet traffic exiting the network. Both *Ingress* and *Egress* filtering are based on identifying addresses within defined IP ranges. Both mechanisms must know the expected range of IP addresses to filter the spoofed traffic at a port effectively. This might be a severe challenge in some networks with complicated topologies. *A reverse path filtering* approach [18] can be applied to *Ingress/Egress filtering* to build such a knowledge base.

*Router-Based Packet Filtering* (RPF) [27] proposed a filtering mechanism with the ability to filter a large portion of spoofed IP traffic at a given time and prevent attack traffics from reaching its targets. Peng et al. [58] analyzed the router-based packet filtering as an upgraded extension of *ingress filtering* to prevent the scope of DDoS attacks from extending into the core of the internet. The difference between RPF and *Ingress filtering* is that RPF uses the route information of the packets to filter out the packets with the spoofed source address. RPF also faces the challenge of network topology as the internet route often changes, which makes real-time updates difficult.

*History-Based IP Filtering.* It was proposed by Peng et al. [57] as a filtering mechanism for preventing DDoS attacks. The underlying idea behind this approach is to filter the IP packets by using an IP database pre-built by the network edge router. This historical information in the database is gathered from previous connections that were established at the edge router. It might be challenging to differentiate between legitimate traffic and attack traffic if the IP traffic is not in the dataset because the edge router has no connection history. An attacker can also attempt to manipulate the system/server to include the IP traffic in the dataset. Douligeris and Mitrokotsa [11] recommends a filtering rule to increase the window an IP address must establish a connection to be considered legitimate traffic.

**Other prevention approaches**. There are a variety of actions that can be implemented in a prevention technique highlighted below,

*Disabling Unused Services.* Geng and Whinston [60] proposed another prevention approach for DDoS attacks. The idea behind the approach is to disable network services that are not in use or needed in a session to avoid exposing them to attacks. If services like UDP echo or character generator services are not in use, the chances of attacks disrupting the services are extremely low.

*Applying Security Patches.* Security patches are known to fix bugs and explorable vulnerabilities in a system to fortify the resistance in a host against malicious threats like DDoS attacks [60]. Updates are essential to any system, and it is essential that host computers carry out updates unassisted, using the latest security patches. These patches could fix bugs and issues with the previous version or deploy the latest prevention techniques against DDoS attacks [11].

*Disabling IP Broadcasts.* Disabling IP broadcasts are to ensure those host computers are not used as amplifiers in ICMP Flood and Smurf attacks anymore. However, for this to be effective as a defence technique, all neighboring IP broadcasts would have to be disabled.

*Changing IP address.* This scheme was introduced as a prevention technique by manipulating the location of the victim's computer [60]. This is achieved by changing the victim's IP address to a new one, thereby making the previous location invalid. When the IP address is changed, the Internet routers and the edge routers will be informed, and illegitimate packets will be dropped.

*Load Balancing.* It is a prevention technique that eliminates the chances of a network disconnecting or shutting out during an attack by providing additional bandwidth over and above what is required. Specht and Lee [54] proposed a failsafe protection scheme to duplicate the servers should there be a loss of resources/connection during an attack.

*Honeypots.* Are systems setup as a diversion from the main target with limited security to attract an attacker to attack the honey pot instead of the main system. This approach was introduced by Weiler [39].

## *2.3 DDoS Attack Detection*

It is essential to detect an attack or a malicious threat as soon as possible. Detection of an attack at an early stage might help safeguard the system resources when adequate security measures are in place. On the other hand, some malicious threats could go unnoticed, resulting in a high impact/damage to a network or system resources. It is, however, important to note that early detection may be based on limited information, and some legitimate behaviours can be mistaken for a threat or an attack. Therefore, an efficient detection mechanism must distinguish malicious packets from legitimate ones with high accuracy, minimal resource consumption, and low false positive and negative rates. Attack detection mechanisms range from simple packet monitoring to sophisticated machine learning-based packet filtering approaches.

**Monitoring Packet Rate**. In DDoS attacks, there is usually a high traffic rate. However, some attack traffic can be mistaken for legitimate traffic and vice versa. Therefore, detection schemes need to be designed to counter the risk of *false positives* during detection [58]. One such example is a scheme called MULTOPS [59], which is used to detect attacks by observing the packet rate to and from the victim. The scheme records the packet rate statistics for traffic flows between hosts. A notable difference between the packet rate to or from a host will indicate an attack, by which MULTOPS can identify the victim and the source of the attack.

**Signature and Misuse Detection**. Signature based detection detects an attack when the observed network traffic matches a known pattern of malicious activity [58]. Signature-based deception can only detect a known malicious threat. Bakr et al. [2] classifies signature-based detection as a misuse detection. Such a detection technique cannot efficiently defend against DDoS attacks.

**Anomaly Based Detection**. On the other hand, Anomaly-based detection is designed to detect and analyze attacks that are unknown and unpredictable. When an observed traffic pattern does not match the behaviour of a normal traffic pattern, an anomaly-based detection system is designed using a set of training data to detect the abnormal traffic pattern. Anomaly based detection can use simple statistical approaches, while more advanced approaches are often based on machine learning technology.

An example statistical scheme is proposed by Blazek et al. [47], a batch detection method with low technical complexity to detect attacks by observing statistical changes in the traffic rate. The detection assumes that, when an attack occurs, there is a noticeable change in traffic patterns. More sophisticated approaches include those proposed by Cheng et al. [44] and Kulkarni and Bush [3], which uses the Kolmogorov algorithm. These approaches with high technical complexity are based on the assumption that multiple attack sources might use the same attack tool.

**Machine Learning Approaches**. Machine learning is an anomaly-based approach that uses a set of trained/training data to analyze, understand and predict an event. The significance of machine learning is the development of systems that can automatically learn from data and analyze them without being instructed to do so. Machine learning techniques can be classified into: Supervised Learning, Semi-Supervised Learning, and Unsupervised Learning.

*In supervised learning*. The set of training data uses supervised learning with labelled data. A few examples of learning algorithms include: *the support Vector Machine (SVM), Hidden Markov Model (HMM), Bayesian Statistics Artificial Neural Networks (ANNs)* [45].

SVM: Sultana et al. [38] is commonly used in network intrusion detection systems because of its strong classification and functionality in computation. SVM is efficient for high dimensional data.

*Unsupervised Learning*. Unsupervised machine learning aims to model the structure or distribution in the data to analyze and understand the data without supervision. With unsupervised learning, there are no incorrect answers or teachers/controllers. The algorithms are meant to identify and structure the data [46].

*Semi-Supervised Learning*. It is a combination of supervised and unsupervised learning that uses labelled data as well as unlabeled data.

*Deep Learning (DL)*. Enables an algorithm to understand the representation of data with different ways of generalization like images, sound and text [28]. Deep learning can be a trained and supervised data learning technique like a convolutional neural network (CNN), the benchmark model for computer vision purposes. CNN approach was implemented by Hussain et al. [8] for early detection of DDoS attacks in a Cyber-physical system (CPS). This was modelled for a 4G LTE architecture. DL can also be unsupervised training using an auto-encoder [28] to learn the encoding for data batch for size reduction.

*Auto-encoder neural network.* This was proposed by Luo and Nagarajany [56] for anomaly detection in wireless sensor networks (WSNs) in an IoT environment. This model places more computationally intensive learning tasks in the cloud as leverage for low computation load on the sensors. A deep belief network (DBN) is another unsupervised DL technique with the ability to reconstruct its input when trained in an unsupervised way with examples. DBNs can also be trained further in a supervised manner to enable classification. Zhou and Paffenroth [12] expressed concern for the lack of clean training data sets and outliers. They introduced a deep auto-encoder scheme (Robust deep auto-encoder) that maintains the ability to discover high-quality features, focusing on eliminating outliers and noise to create clean training data sets.

Other types of deep learning models include Vanilla Deep Neural Network (DNN), Self-taught learning (STL) and Recurrent Neutral network (RNN) [45]. RNN can either be a supervised or unsupervised learning method.

## *2.4 DDOS Attack Mitigation Schemes*

Mitigation measures are actions taken during an attack to reduce the impact of an attack on a network system. DDoS attack Mitigation generally occurs during or after an attack. Some prevention techniques are also considered mitigation measures. There is no 100% guarantee in protecting against a DDoS attack; thus, the solutions proposed are to mitigate the effects of a DDoS attack on a system.

Several mitigation schemes have been proposed to reduce the effects of a DDoS attack over the years. In this section, we will be discussing the proposed schemes.

Functional elements in mitigation are the critical actions taken to mitigate an attack by reducing the attack's impact or preserving the network resource. Which in most cases are drop traffic, block, or allow flow, redirect flow paths for DPI, deploy mitigation agents, and more sophisticated approaches involving the deployment of honeypots, virtualized network function for high-level detection based on threat classification, dynamically changing the IP address (of a target under attack).

**IP Traceback**. A manual traceback approach, where the network administrator of a network under attack can call his Internet service provider for information on the source or path of the packets, would be very tedious. Thus various proposals have surfaced to automate this process in the past [11]. To do this, multiple packets are required from each hop to reconstruct the path of attack traffic.

Traceback techniques are used as solutions in locating the original source of a DDoS attack, considering most attackers spoof the source address [2].

IP Traceback technique can trace the attacking traffic back to its source by utilizing the routers in the path with path characterization [58]. Backscatter is an IP traceback scheme proposed by Gemberling et al. [9].

Burch and Cheswick [23] proposed a Link-Testing Traceback Technique that enables the host to observe and test its incoming links as a likely input link for attack traffic. This scheme deduces the path of the attack by flooding the links with large bursts of traffic and then analyzing how it affects the network.

*Probabilistic IP Traceback Schemes.* The concept of the probabilistic IP traceback is that the routers are capable of including partial path information into incoming traffic to enable the target router to reconstruct the packet path using this information [58].

*ICMP traceback.* A traceback mechanism proposed by Bellovin et al. [53] ensures that every router analyses the outgoing packets with a low probability (1 out of 20,000) before forwarding the ICMP traceback message to the destination [11]. The ICMP traceback message generated is known as the iTrace packet, which consists of the sending router's address [58].

**Packet Marking**. The methods of traceback could be supported by packet marking. Packet marking is a tracking method where routers generate a unique random number to mark packets received to compare with the threshold value [61]. The standard packet marking techniques are probabilistic packet marking (PPM) and deterministic packet marking (DPM).

*Probabilistic packet marking (PPM).* First introduced by Savage et al. [52] has received a great deal of traction in the research community. Over the years, this technique has been iteratively improved to counter novel DDoS attacks. The underlying idea behind PPM is that each router marks the packets while they travel between the source and the destination by embedding its IP address (path information) to the packets. PPM has been upgraded over the years to make it sophisticated enough to adapt to different types of DDoS attacks. Song and Perrig [17] enhanced the functionalities of PPM in efficiency and security by proposing a *hashing algorithm* to encode the embedded path information for authenticating the routers. Dean et al. [16], introduced a coding scheme with an algebraic approach to attach the path information to the packets. Unlike previous marking schemes, this scheme does not require many packets to reconstruct the attack path. However, it is less efficient in a scenario involving multiple attackers. Park and Lee (2004) proposed a distribution filter for the routers to enable the routers to filter the incoming packets according to the network topology. This scheme can help mitigate spoofed traffic at an early stage.

*Deterministic packet marking.* The router marks every incoming packet with a unique identifier in this approach. This scheme can overwrite any spoofed packets with correct marks. Compared with PPM, DPM requires less overhead and computation and is faster with minimal false-positive rates [2].

**Hybrid Approaches**. Bakr et al. [2] introduced an approach, "Flexible, collaborative, multilayer, DDoS Prevention Framework (FCMDPF)" developed by Saleh and Abdul Manaf [33], which controls the categories of HTTP-based distributed denial of service attack via three framework layers. The first layer is a prevention framework that contains an IP blacklist table. This layer blocks the attacking IP source if it matches any address in the blacklist table.

The second layer is the service traceback oriented architecture (STBOA) used to detect the source of the traffic, mainly if it is human-generated or a botnet.

The final layer of the framework is an entropy-based scheme designed to discard packets that belong to high-rate DDoS traffic flows and flash crowd attacks.

# 3 5G/IoT Security Solutions Against DDoS Attack

Earlier, we discussed the three categories of defense mechanisms against DDoS. This section analyses security approaches proposed to combat DDoS attacks under these three categories in 5G/IoT. One of the most common approaches for security solutions used in most network environments is software-defined network frameworks.

The software-defined network (SDN) framework offers a centralized control approach used in threat detection and attack mitigation by monitoring the entire network [22]. SDN can be deployed with a dynamic approach [55]. An SDN-based approach is used to provide security resilience, continuous monitoring, and adaptive decision-making in an IoT topology in a dynamic and dynamic and adaptive manner. In most SDN frameworks, there are SDN-enabled switches, SDN controllers and in this case, SDN master controllers and IoT devices. However, SDN-based security approaches in 5G models have not yet gained much traction like other 5G related techniques such as IoT.

## *3.1 DDoS Attack Prevention for 5G/IoT*

As stated earlier, prevention is key to avoiding an attack reaching or penetrating the network defense. Iavich et al. [31] suggested an integration of cyber security modules on every 5G station as an additional server. The server will consist of the firewall and IDS/IPS system. This will not be a cost-effective approach as it will be expensive to deploy security solutions at each and every individual station separately. An SDN approach discussed in [55], uses a simple hierarchical approach where the switches monitor the traffic flows coming from the IoT devices in a cluster while mirroring a traceback feature to the SDN controller of the device cluster. The cluster controller can analyze the packets to detect malicious behaviours using a learning model based on the historical statistics of such traffic profiles. The classification model uses machine learning to classify the traffic as either malicious or legitimate before the flow management can generate rules or actions for the switch to perform on the incoming traffic. In this scheme, the cluster controller sends an update to the SDN master controller. The model could have adopted a peer-to-peer approach between the controllers to generate some form of communication between clusters or, better still, configured the master controller to propagate the updates to other controllers about a particular node in another cluster, and should the node eventually relocate to a new cluster the cluster SDN controller would already have the statistics about this node. Dao et al. [40] researched the multi-access edge computing technologies to develop an edge prevention mechanism in 5G called MAEC-X. This model consists of a MAEC-X controller in the core network to collect attack data from clients for analysis, broadcast identified attack warnings to all clients, and generate action policies based on the edge node participating in the attack.

## *3.2 DDoS Attack Detection for 5G/IoT*

Detection is an essential part of most security solutions in 5G. It is important to know when an attack was generated and from where, as in the case of the proposed source side detection scheme for DDoS in IoT enabled 5G environment [34]. Source-side inspection observes discordant behaviours where traffic flows are inspected at the source-side by delegating an analytic task to a specific data processing layer, where advanced feature extraction, pattern recognition, prediction and adaptive thresholding capabilities operate.

In an SDN environment, the flow statistics have proven vital to the analysis of a potential attack. Kalliola et al. [1] designed an approach to combat DDoS flooding attacks in an autonomous system. This scheme was largely automated with traffic learning and elastic control invocations like load control and filtering. They proposed to use an external blacklist data from a third-party intrusion detection system (IDS). This can always be manipulated by an attacker. The flow-oriented framework was based on a traffic allocation and control scheme which checks the volume ratio of normal traffic against attack traffic profiles in a traffic cluster (The cluster with the best ratio of normal to attack traffic has the most normal traffic during an attack).

Just like the source side, 5G can consist of more segments of the edge and the network's core. Edge computing is considered one of the crucial emerging technologies for computer network functionalities. As discussed in the previous section, MAEC-X is a technology researched by Dao et al. [40]. The model also has several MAEC-X clients at the edge, which are installed with an attack detection module that monitors the traffic in real-time and collects suspicious user traffic behaviours and characteristics. Bhardwaj et al. [26] proposed the ShadowNet idea to increase the detection and response speed in an IoT-DDoS attack. Serrano Mamolar et al. [4] introduced the architecture of a mobile edge 5G multi-tenant infrastructure with the edge, the core and the inter-domain structure in the network. The mitigation design framework is based on the Snort IDS capabilities named the snort monitoring agent (SMA). Mamolar et al. [6] had the third segment as a multi-domain structure.

An IoT defence approach using SDN edge defence was introduced in [32], where SDN and fog computing are combined together to detect and mitigate IoT device bot attacks at the edge of the network. There are two algorithms in [30], with the first being a threshold random walk with credit-based rate limiting. This stores a queue of the TCP SYNs for every connected device to enable an efficient scan of the connection attempts against a successful connection rate. In contrast, the second algorithm is the rate-limiting scheme used to detect malicious nodes by observing the nodes attempting multiple connections in a short period. And this is done by checking new connection requests against the recently connected host.

### *3.3 DDoS Attack Mitigation for 5G/IoT*

Mitigation is the third form of defence against DDoS attacks. It is an action to be taken to reduce or stop the effects of an attack. This action can be executed in any segment of a 5G network, even in a network slice. Network slicing is one of the significant technologies in 5G networks that provides flexibility and scalability and can provide security [62], where multiple network services are hosted on the same physical network resource utilizing the virtualized infrastructures. This offers dedicated service to each slice and a slice can be an industry or a service. A DDoS attack on a slice can affect the performance of other services as they might share the same physical resources. Sattar and Matrawy [15] proposed a mitigation scheme using slice isolation to provide inter-slice and intra-slice isolation and increase the availability of a slice in a DDoS attack. Inter-slice isolation provides mitigation against DDoS attacks because the hardware resources are not shared between the slices and DDoS on one slice does not impact the other slices. At the same time, intra-slice isolation provides better availability for the slices since the components of the slice are hosted on different hosts.

5G segments, as discussed in the previous section, can be grouped into multi-tenant networks, which might include the edge, fog, cloud, and core of the network. A multi-tenant approach consisting of the edge and core network environment was proposed in [5], where it was an improved SMA technology from [4] integrated with unified2 [49] standard format. In this model, there are three levels, user level, tenant level and flow level. The network flow controller was developed to act at any point of the data path as the extended IDS can detect malicious flows for multi-tenancy and user mobility. To achieve a self-managed model, the mitigation architecture was sectioned into categories:

- SMA: uses snort IDS and unified2.
- Decision Maker: where the administrators automate the decision-making task based on a set of rules,
- Action Enforcer: actions what to do directly to the system and translates the decisions to be compatible with the network topology.

- Flow control agent: Designed to allow distributed mitigation and to be API compatible with different implementations of data path.

Some attackers are sophisticated enough to generate random and unpredictable attacks. Hong et al. [20] proposed a threshold-based approach as an entropy-based mitigation scheme in SDN using Open-Day-Light (control all switches). The proposed mechanism uses the information collected from traffic status to calculate the entropy of the network environment. It is configured to prevent misjudgment if the entropy of the system slowly approaches a steady state over time. In the event that there is no attack, the mitigation scheme balances the load of the network devices like switches and servers. However, if the system is not under attack, the mitigation approach will continue to run and consume more bandwidth.

Some flow-based mitigation approaches operate by combining OpenFlow controllers and Flow monitoring engines [13, 25]. Buragohain and Medhi [13] uses OpenFlow for executing mitigation rules in the switches. With detection and mitigation in a data centre environment, the scheme FlowTrApp uses a flow rate and flow duration algorithm to detect legitimate traffic based on the number of benign traffic flows a node generates and for how long. It also collects statistics from legitimate traffic as a traffic flow tuple which includes minimum and maximum values for both traffic rate and traffic duration as a threshold. A parameter is also set in layer 7 to restrict HTTP applications to a single session at a given time.

## *3.4 Hybrid Approach Against DDoS for 5G/IoT*

We categorize approaches that consist of 2–3 of the security solutions of prevention, detection, and mitigation as a hybrid approach. Due to the hierarchical nature of SDN, you will find most hybrid approaches in SDN for most network environments as the SDN centralized controller can be responsible for making mitigation decisions while prevention and detection might be deployed at a different section of the network. An SDN 5G-oriented solution was proposed by Perez et al. [35] as a combination with network function virtualization (NFV). The design included a high-level detection which analysis network flows to identify a suspicious node/bot. The nodes are deployed with mobility capabilities which raises a tracking concern of malicious nodes. After the analysis, it uses a deep packet inspection mechanism for confirmation. The solution uses a virtualized honeynet as a mitigation solution once the deep packet inspection mechanism confirms the presence of an attack. Giotis et al. [25] introduced an anomaly detection approach by extending the functionalities of the OpenFlow operator to ensure scalability and efficiency. The three branches of the architecture are a collector (flow statistics collection), anomaly detection (flow statistics analysis, anomaly detection and identification) and anomaly mitigation (whitelist function, anomaly mitigation). In [43] the three stages of the tenancy are the edge, the fog and the cloud computing structure. The network's edge computing stage/level is in the network layer between the nodes and the network. The solutions are mainly IoT SDN gateways with firmware, light intrusion detection or vulnerability scanning systems. Fog computing is the IoT control unit that includes a cluster of SDN controllers which collects, analyzes traffic data, detects malicious traffic, and provides a mitigation solution. Having mitigation approaches for the stages in a multi-tenant network provides efficient solutions for the network. But there is an issue of scalability in a DDoS attack. Including an NFV approach in such a solution would solve scalability issues.

## 4 Conclusion

The seamless connectivity in a 5G network can also create vulnerability in the network for attacks such as distributed denial of service (DDoS), as attackers might have easy access to the network infrastructure. There are several proposed methods in all three categories for prevention, detection, and mitigation, and some models implement all three categories to combat DDoS attacks on different network environments. However, not so much has been proposed in recent years for 5G. We cannot run all the proposed approaches from each category at the same time because it might be difficult, cumbersome, and highly expensive to run. Cost is a factor considered by many organizations in network infrastructure setup. Developing an efficient model which could also be cost-effective is an essential objective in any network system. In a network security solution, it is very expensive to have all possible security functions, methods, and models operational at all times. Costs, including operational cost, network resource usage, and power usage, will be very high. Thus, there is a need for a lightweight yet scalable, efficient approach.

A cost effective and highly efficient security model should be flexible and reactive to any attack scenario or network circumstances when needed. Deploying a security policy language that is reactive and event-oriented fits into a flexible, efficient, and lightweight security approach. A policy is a set of rules or actions defined as conditions to enforce specific mitigation functions [29]. Introducing a policy management scheme to deploy the appropriate mitigation mechanism to a specific threat at the required time will be effective as a security solution for 5G technology to DDoS attacks to manage the response to threats posed by DDoS attacks.

## References

1. A. Kalliola, et al., Flooding DDoS mitigation and traffic management with software defined networking, in *2015 IEEE 4th International Conference on Cloud Networking, CloudNet 2015* (2015), pp. 248–254. https://doi.org/10.1109/CloudNet.2015.7335317
2. A. Bakr, A. El-Aziz, H. Hefny, A survey on mitigation techniques against DDoS attacks on cloud computing architecture. Int. J. Adv. Sci. Technol.**28**(12), 187–200 (2019). http://sersc.org/journals/index.php/IJAST/article/view/1211/994

3. A. Kulkarni, S. Bush, Detecting distributed denial-of-service attacks using Kolmogorov complexity metrics. J. Netw. Syst. Manag. **14**(1), 69–80 (2006). https://doi.org/10.1007/s10922-005-9016-3
4. A. Serrano Mamolar, et al., Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks. Comput. Secur. **79**(May 2020), 132–147 (2018). https://doi.org/10.1016/j.cose.2018.07.017
5. A. Serrano Mamolar, et al., Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks. J. Netw. Comput. Appl. **145**(November 2018), 102416 (Elsevier Ltd, 2019). https://doi.org/10.1016/j.jnca.2019.102416
6. A.S. Mamolar, et al., Towards the detection of mobile DDoS attacks in 5G multi-tenant networks, in *2019 European Conference on Networks and Communications, EuCNC 2019* (IEEE, 2019), pp. 273–277. https://doi.org/10.1109/EuCNC.2019.8801975
7. B. Gwak, et al., IoT trust estimation in an unknown place using the opinions of i-sharing friends, in *Proceedings—16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 11th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Conference on Embedded Software and Systems* (2017), pp. 602–609. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.290
8. B. Hussain, et al., Deep learning-based DDoS-attack detection for cyber-physical system over 5G network. IEEE Trans. Ind. Inf. **3203**(DL), 1–1 (2020). https://doi.org/10.1109/tii.2020.2974520
9. B.W. Gemberling, C.L. Morrow, B.R. Greene, ISP security—Real world techniques, in *NANOG* (2001).
10. C. Douligeris, A. Mitrokotsa, DDoS attacks and defense mechanisms: a classification, in *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2003* (2003), pp. 190–193. https://doi.org/10.1109/ISSPIT.2003.1341092
11. C. Douligeris, A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput. Netw. **44**(5), 643–666 (2004). https://doi.org/10.1016/j.comnet.2003.10.003
12. C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1296 (2017). pp. 665–674. https://doi.org/10.1145/3097983.3098052
13. C. Buragohain, N. Medhi, FlowTrApp: an SDN based architecture for DDoS attack detection and mitigation in data centers, in *3rd International Conference on Signal Processing and Integrated Networks, SPIN 2016* (IEEE, 2016), pp. 519–524. https://doi.org/10.1109/SPIN.2016.7566750
14. Chiang et al., Fog and IoT : an overview of research opportunities. IEEE Internet Things J. **3**(6), 854–864 (IEEE, 2016). https://doi.org/10.1109/JIOT.2016.2584538
15. D. Sattar, A. Matrawy, Towards secure slicing: using slice isolation to mitigate DDoS attacks on 5G core network slices, in *2019 IEEE Conference on Communications and Network Security, CNS 2019* (2019), pp. 82–90. https://doi.org/10.1109/CNS.2019.8802852
16. D. Dean, M. Franklin, A. Stubblefield, An algebraic approach to IP traceback. ACM Transactions on Information and System Security **5**(2), pp. 119–137 (2002). https://doi.org/10.1145/505586.505588
17. D.X. Song, A. Perrig, Advanced and authenticated marking schemes for IP traceback, Proceedings—IEEE INFOCOM, **2**, (2001). pp. 878–886. https://doi.org/10.1109/INFCOM.2001.916279
18. F. Baker, Requirements for IP version 4 routers. IETF, RFC 1812 (1995).
19. F. Wong, C.X. Tan, A survey of trends in massive DDOS attacks and cloud-based mitigations. Int. J. Netw. Secur. Appl. **6**(3), 57–71 (2014). https://doi.org/10.5121/ijnsa.2014.6305
20. G.C. Hong, C.N. Lee, M.F. Lee, Dynamic threshold for DDoS mitigation in SDN environment, in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019* (IEEE, 2019), pp. 1–7. https://doi.org/10.1109/APSIPAASC47483.2019.9023229

21. H. Ghorbani, M.S. Mohammadzadeh, M.H. Ahmadzadegan, DDoS attacks on the IoT network with the emergence of 5G, in *2020 International Conference on Technology and Entrepreneurship—Virtual, ICTE-V 2020* (2020). https://doi.org/10.1109/ICTE-V50708.2020.9113779
22. H. Huang, J. Chu, X. Cheng, Trend analysis and countermeasure research of DDoS attack under 5G network, in *2021 IEEE 5th International Conference on Cryptography, Security and Privacy, CSP 2021* (no. 978, 2021), pp. 153–160. https://doi.org/10.1109/CSP51677.2021.9357499
23. H. Burch, B. Cheswick, Tracing anonymous packets to their approximate source, Lisa (2000)
24. J. Rodriguez, *Fundamentals of 5G Mobile Networks*, *Fundamentals of 5G Mobile Networks* (2015). https://doi.org/10.1002/9781118867464
25. K. Giotis et al., Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Comput. Netw. **62**(April), 122–136 (2014). https://doi.org/10.1016/j.bjp.2013.10.014
26. K. Bhardwaj, J.C. Miranda, A. Gavrilovska, Towards IoT-DDoS prevention using edge computing, in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)* (2018). https://www.usenix.org/biblio-1765
27. K. Park, H. Lee, On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack, Proceedings—IEEE INFOCOM, **1**, pp. 338–347. (2000) https://doi.org/10.1109/INFCOM.2001.916716
28. L. Deng, D. Yu, Deep learning: methods and applications. Found. Trends Signal Process. **7**(3–4), 197–387 (2013). https://doi.org/10.1561/2000000039
29. L. Kagal, T. Finin, A. Joshi, A policy language for a pervasive computing environment, in *Proceedings—POLICY 2003: IEEE 4th International Workshop on Policies for Distributed Systems and Networks* (2003), pp. 63–74. https://doi.org/10.1109/POLICY.2003.1206958
30. M. Ejaz Ahmed, H. Kim, DDoS attack mitigation in internet of things using software defined networking, in *Proceedings—3rd IEEE International Conference on Big Data Computing Service and Applications, BigDataService 2017* (2017), pp. 271–276. https://doi.org/10.1109/BigDataService.2017.41
31. M. Iavich, et al., The novel system of attacks detection in 5G. Lect. Notes Netw. Syst. **226** LNNS(April), 580–591 (2021). https://doi.org/10.1007/978-3-030-75075-6_47
32. M. Ozcelik, N. Chalabianloo, G. Gur, Software-defined edge defense against IoT-based DDoS, in *IEEE CIT 2017—17th IEEE International Conference on Computer and Information Technology* (2017), pp. 308–313. https://doi.org/10.1109/CIT.2017.61
33. M.A. Saleh, S. Abdul Manaf, A novel protective framework for defeating HTTP-based denial of service and distributed denial of service attacks. Sci. World J. (2015). https://doi.org/10.1155/2015/238230
34. M.A. Sotelo Monge, et al., Source-side DDoS detection on IoT-enabled 5G environments, in *Proceedings—2018 International Workshop on Secure Internet of Things, SIoT 2018* (2018), pp. 28–37. https://doi.org/10.1109/SIoT.2018.00010
35. M.G. Perez et al., Dynamic reconfiguration in 5G mobile networks to proactively detect and mitigate botnets. IEEE Internet Comput. **21**(5), 28–36 (2017). https://doi.org/10.1109/MIC.2017.3481345
36. N. Jawad, et al., Smart television services using NFV/SDN network management. IEEE Trans. Broadcast. **65**(2), 404–413 (IEEE, 2019). https://doi.org/10.1109/TBC.2019.2898159
37. N. Patani, R. Patel, A mechanism for prevention of flooding based DDoS attack. Int. J. Comput. Intell. Res. **13**(1), 101–111 (2017)
38. N. Sultana, et al., Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Netw. Appl. **12**(2), 493–501 (2019). https://doi.org/10.1007/s12083-017-0630-0
39. N. Weiler, Honeypots for distributed denial-of-service attacks, in *Proceedings of the Workshop on Enabling Technologies: infrastructure for Collaborative Enterprises, WETICE* (2002). pp. 109–114. https://doi.org/10.1109/ENABL.2002.1029997

40. N.N. Dao, et al., MAEC-X: DDoS prevention leveraging multi-access edge computing, in *International Conference on Information Networking* (IEEE, 2018), pp. 245–248. https://doi.org/10.1109/ICOIN.2018.8343118
41. P. Ferguson, D. Senie, Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. IETF (1998). https://www.hjp.at/doc/rfc/rfc2267.html
42. Q. Gu, S. Marcos, Denial of service attacks department of computer science texas State University—San marcos school of information sciences and technology Pennsylvania State University denial of service attacks outline (2007), pp. 1–28. https://s2.ist.psu.edu/ist451/DDoS-Chap-Gu-June-07.pdf
43. Q. Yan et al., A multi-level DDoS mitigation framework for the industrial internet of things. IEEE Commun. Mag. IEEE **56**(2), 30–36 (2018). https://doi.org/10.1109/MCOM.2018.1700621
44. Q. Cheng, F. Fang, Kolmogorov random graphs only have trivial stable colorings. Information processing letters, **81**(3), (2001). 133–136 https://doi.org/10.1016/S0304-3975(96)00206-X
45. R. Alhajri, R. Zagrouba, F. Al-Haidari, Survey for anomaly detection of IoT botnets using machine learning auto-encoders. Int. J. Appl. Eng. Res.**14**(10), 2417–2421 (2019). http://www.ripublication.com
46. R. Sathya, A. Abraham, Comparison of supervised and unsupervised learning algorithms for pattern classification. (IJARAI) Int. J. Adv. Res. Artific. Intell. **2**(2), 34–38 (2013)
47. R.B. Blazek, et al., A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods, in *Proceedings of IEEE* …(2001), pp. 1–7. http://www.professores.unirg.edu.br/marcelo/coordenacao/mar/doutorado/ufrj/DoSDectionPaper.pdf
48. RCN, *Prevention is Better than Cure*, *Royal College of Nursing* (2020). https://www.rcn.org.uk/get-involved/campaign-with-us/prevention-is-better-than-cure
49. README.unified2, *Unified2* (2018). https://www.snort.org/faq/readme-unified2. Accessed 15 Apr. 2022
50. S. Li, L.D. Xu, S. Zhao, 5G Internet of things: a survey. J. Ind. Inform. Integr. **10**, 1–9 (Elsevier, 2018). https://doi.org/10.1016/J.JII.2018.01.005
51. S. Rommer et al., *5G Core Networks* (Elsevier, 2020)
52. S. Savage, D. Wetherall, A. Karlin, T. Anderson, Network support for IP traceback. IEEE/ACM Transactions on Networking **9**(3), 226–237 (2000). https://doi.org/10.1109/90.929847
53. S.M. Bellovin, M. Leech, T. Taylor, ICMP traceback messages (2003). http://academiccommons.columbia.edu/catalog/ac:127253
54. S.M. Specht, R.B. Lee, Distributed denial of service: taxonomies of attacks, tools and countermeasures, Int. Works. Secur. Parall. Distrib. Syst. (9), 543–550 (2004). https://doi.org/10.1.1.133.4566
55. S.S. Bhunia, M. Gurusamy, Dynamic attack detection and mitigation in IoT using SDN, in *2017 27th International Telecommunication Networks and Applications Conference, ITNAC 2017* (2017), pp. 1–6. https://doi.org/10.1109/ATNAC.2017.8215418
56. T. Luo, S.G. Nagarajany, Distributed anomaly detection using autoencoder neural networks in WSN for IoT, in *IEEE International Conference on Communications* (2018). https://doi.org/10.1109/ICC.2018.8422402
57. T. Peng, C. Leckie, K. Ramamohanarao, Protection from distributed denial of service attack using history-based IP filtering (2002)
58. T. Peng, C. Leckie, K. Ramamohanarao, Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv. **39**(1) (2007). https://doi.org/10.1145/1216370.1216373
59. T.M. Gil, M. Poletto, in *Proceedings of the 10 th USENIX Security Symposium MULTOPS : a Data-Structure for Bandwidth Attack Detection. Statistics* (2001)
60. X. Geng, A.B. Whinston, Proactively defeating distributed denial of service attacks, *Security* **1**(August), 1520 (2000). https://doi.org/10.1109/ccece.2003.1226075
61. Y. Bhavani, V. Janaki, R. Sridevi, IP traceback through modified probabilistic packet marking algorithm using chinese remainder theorem. Ain Shams Eng. J. Faculty Eng. Ain Shams Univ. **6**(2), 715–722 (2015). https://doi.org/10.1016/j.asej.2014.12.004

62. Z. Kotulski, et al., On end-to-end approach for slice isolation in 5G networks. Fundamental challenges, in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017* (vol. 11, 2017), pp. 783–792. https://doi.org/10.15439/2017F228

# A Lightweight Blockchain-Based Trust Management Framework for Access Control in IoT

**Tianyu Zhao, Ernest Foo, and Hui Tian**

**Abstract** Blockchain technology has provided lots of significant benefits in terms of security, auditability, immutability and anonymity. Following on from these remarkable features, blockchain technology has been incorporated with lots of non-monetary applications including the Internet of Things (IoT). On the other hand, blockchain can be used for trust management for IoT. However, a major challenge is to find an appropriate light weight consensus algorithm that can be implemented in IoT devices, which have suffered limited computational resources. Building upon the idea of using blockchain as the basic framework, this chapter proposes a lightweight blockchain-based trust management framework that is suitable for IoT devices. Our framework is built upon high resource devices to form the underlying Peer-to-Peer (P2P) network. In addition, we use the smart contract mechanism to generate a trustworthy environment for IoT devices. With the trust evaluation approach, we propose a reputation-based consensus algorithm which can significantly decrease the mining time. Moreover, the verification mechanism can incorporate with the reputation approach to reduce the processing time of block verification. Simulations have demonstrated that our framework achieves low delay time, high Transactions Per Second (TPS) and less processing time compared with relevant baselines. More importantly, our framework shows that it is resilient to several security attacks in blockchain systems.

**Keywords** Blockchain · IoT · Access control · Consensus algorithm

T. Zhao (✉) · E. Foo · H. Tian
School of Information and Communication Technology, Griffith University, Brisbane 4111, Australia
e-mail: tianyu.zhao@griffithuni.edu.au

E. Foo
e-mail: e.foo@griffith.edu.au

H. Tian
e-mail: hui.tian@griffithuni.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_6

# 1 Introduction

In this section, we focus on providing background information on IoT, the access control system and blockchain technology. Furthermore, we discuss the current limitations of implementing blockchain with the access control system in the IoT environment. Lastly, we present our contributions in this chapter.

## *1.1 Overview of IoT*

There is a growing number of devices connected to the Internet due to the rapid development of communication and networking technologies. The connection of devices has led to the development of the Internet of Things (IoT). IoT is a system that integrated physical objects with the digital world, which can collect and share data through the Internet [1]. Therefore, there is a large amount applications based on the IoT framework, such as, smart city, intelligent transportation, smart grid, etc. [2]. According to a recent Gartner report, there will be 20.4 billion IoT devices in 2022 [3]. From a study in [4], it clearly stated that the size of the IoT industry will increase to 70 billion devices in 2025.

Since the IoT systems are different from the traditional systems, there are some unique features needed to be considered. For an IoT architecture, the devices are suffered from resource constrained, which include memory, bandwidth and computation power. Moreover, the devices can connect and disconnect from the network at any time. Therefore, the security mechanism needs to achieve the same requirements based on the properties of IoT systems, such as, delay time, scalability, and packet overhead.

## *1.2 Overview of an Access Control Mechanism*

Access control is a security mechanism that can grant or deny a selective restriction of access to specific users based on who they are and what they are looking for [5]. That is, it ensures the authentication and authorization processes. In an access control mechanism, there is a set of conditions to determine the access ability of a user to resources. As shown in Fig. 1, the access control model can protect every access to a resource based on defined conditions given by the system. Access control shall ensure resource confidentiality, integrity, and availability in the system. Therefore, a complete and effective access control mechanism must consist of authentication, authorization and accountability [6].

In most of the access control models, there is always an attribute authority (AA), which stores all the information related to devices in the network. In this case, this kind of approach may encounter several challenges and limitations. Firstly, it has a

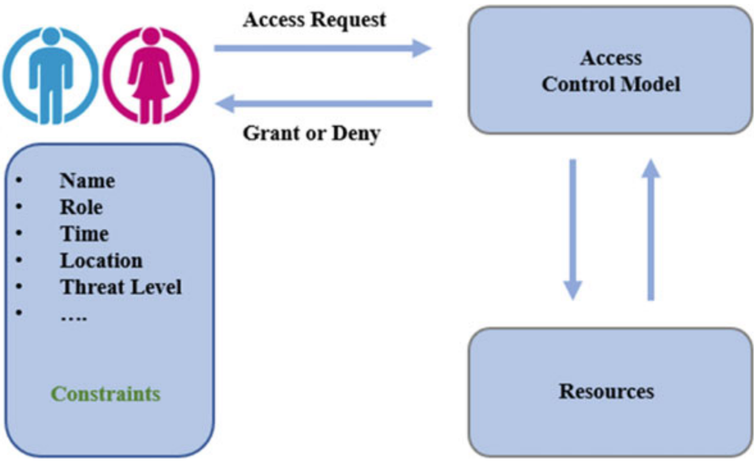


**Fig. 1** The access control mechanism

weakness in scalability as a large amount of IoT devices join to the network. It may cause a burden to the AA and affect its performance. Secondly, it may cause a single point of failure [7]. In the access control architecture, each module is unique and substitutable. The whole system can be affected once a single module is damaged.

## *1.3 Overview of Blockchain*

Blockchain is a distributed ledger that can be used for storing and securely sharing data [8]. As illustrated in Fig. 2, each node operates in a peer-to-peer network and has its own digital ledger. The stored data can be regarded as payment history, such as, Bitcoin, or a contract and perhaps personal data. That information stored in the blockchain provides high auditability for all transactions [9]. New transactions and blocks can be verified by other nodes in the system, thus removing the need for the centralized authority. Therefore, the unique data structure of blockchain can overcome the limitations in the current access control models.

One of the important elements in a blockchain network is the consensus algorithm. The consensus algorithm is a fault-tolerant mechanism that can ensure each node can achieve the necessary agreement about a state of data in a blockchain network. With the support of the consensus algorithm, each party have the same privilege in the blockchain. Therefore, there is no need for a centralization entity to manage all participating nodes. Some of the most representative consensus algorithms may include Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS) and Practical Byzantine Fault Tolerance (BPFT) [10].

Another basic element in blockchain technology is the peer-to-peer (P2P) network. Normally, it builds on a top of an existing network, which can be the Internet. As

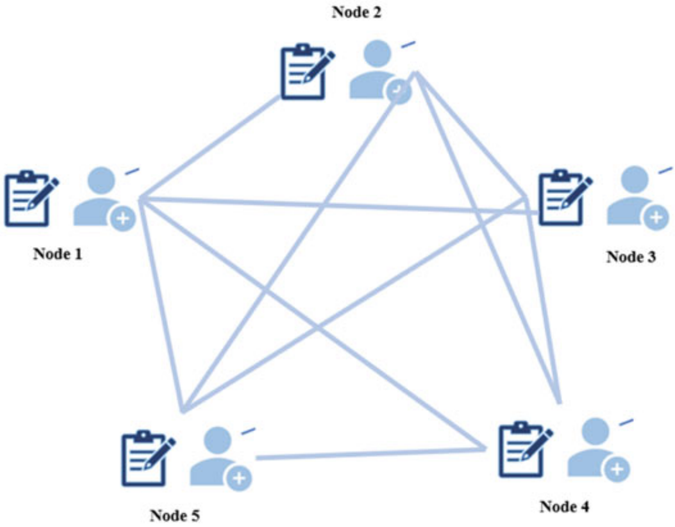
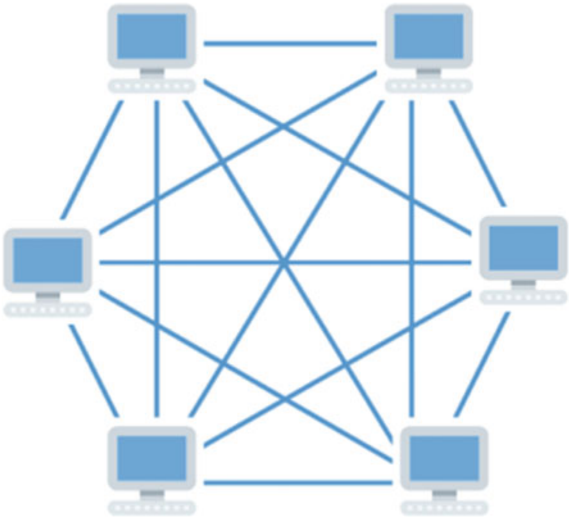


**Fig. 2** The network structure in blockchain

shown in Fig. 3, it presents the way of interaction in the peer-to-peer network. The nodes can find other nodes not by IP address, but by the specific logical identifiers, which are known by all members in the peer-to-peer network [11]. Currently, there are two kinds of routing schemes based on the structure of a network, which include routing in unstructured overlays and routing in structured overlays.

**Fig. 3** The architecture of a peer-to-peer network



Cryptography plays an important role to provide security and privacy in the blockchain. Since each participating node does not trust each other in a distributed network, the purpose of cryptographic schemes is to establish a trust environment. More specifically, in Bitcoin, those schemes can ensure cryptocurrency integrity, confidentiality and availability. Integrity means that unauthorized users cannot modify data in a system. Confidentiality indicates that unauthorized users do not have permission to access data in a system. Moreover, availability means that authorized users can obtain data when they need them. Thus, the cryptographic schemes ensure the security of blockchain technology.

### *1.4 Limitations of Blockchain*

In the context of IoT, a huge amount of data is generated and shared through different networks all time. In this case, the main security concern is the trust management of generated data. From related works and research, the access control mechanism is an efficient tool to protect data in IoT networks. However, the traditional frameworks suffer a few challenges [12]. Firstly, there is limited resource consumption in the IoT network. The performance of IoT devices may be affected when implementing complex data management mechanisms. Second, as mentioned earlier, there are many IoT systems based on a centralized entity to do authentication and authorization. This model does not have a good scalability feature when lots of IoT devices are connected. Currently, blockchain technology can provide lots of benefits for those challenges. To ensure that the blockchain concept is perfectly implemented with assess control in IoT, we need to consider the following limitations:

Limitation 1 (L1): In blockchain systems, it cannot ensure that each node is honest and trustworthy. There may have nodes that deliver unsatisfactory service to other nodes.

Limitation 2 (L2): The existing consensus algorithm needed lots of computational resources, which is far beyond the capabilities of most of IoT devices.

Limitation 3 (L3): In some blockchain systems, a transaction needs a large amount of time to be confirmed by participating nodes. For example, It may take up to 30 min for a new transaction to be confirmed in Bitcoin. However, there is a significant time requirement for most IoT devices.

In order to address the above challenges, distributed trust management needed to be built for access control in IoT systems. By considering the properties of blockchain technology, we need to make some modifications to meet the features of IoT systems. Particularly, Current consensus algorithms have different pitfalls when used in our distributed trust management approach. Therefore, we need to propose a new consensus algorithm, which can achieve our requirements in this chapter. In the meantime, there are some trade-offs needed to be made to ensure our approach achieves the following requirements.

Requirement 1 (R1): All behaviours of the nodes conducted in the blockchain system needs to be evaluated, as we need to ensure honesty and trustworthiness.

Requirement 2 (R2): All the nodes need to reach a consensus without consuming too many computational resources.

Requirement 3 (R3): The consensus algorithm must be able to provide low latency in IoT environment, as most IoT devices have the delay requirement.

## *1.5 Contributions of the Chapter*

In this chapter, by addressing the aforementioned concerns, we propose several contributions to solve related limitations and challenges. To ensure the trustworthiness and honesty of each node, we introduce trust management with smart contracts to evaluate the trust and reputation score in a blockchain network. In smart contracts, we predefined some rules to ensure that each node adheres to the same principles and parameters. We also propose a reputation-based consensus algorithm that is combined with a random sleeping mechanism. In the consensus algorithm, the reputation score of each node needs to be higher than the threshold to become the potential miners. Moreover, each node must take a random sleep time before mining. The proposed consensus algorithm can protect against malicious nodes and mitigate centralization issues at a certain level. Lastly, we propose a reputation-based verification mechanism to randomly verify N% of transactions in the block that are mined from the miner. The higher the reputation score of the miner, the fewer transactions needed to be verified in the blockchain network. The proposed mechanism can boost the speed of the confirmation process for each block in the IoT system. The key contributions of this chapter are summarized below:

1. We propose a trust management blockchain framework for access control, that is tailored to meet the specific requirements in IoT environments. We incorporate several smart contracts to ensure the trustworthy and honest behaviours of each node in this framework. We also improve the existed consensus algorithms to meet the desired requirements in IoT systems.
2. We demonstrate that our proposed framework is securely by taking quantitative analysis from related cyber-attacks. Furthermore, we conduct a risk analysis to investigate potential cyber-attacks that may happen in our framework.
3. We undertake experiments to evaluate the performance of our trust-based consensus and verification algorithm by comparing with other consensus approaches. We demonstrate that our approaches perform well in terms of latency, accuracy and scalability.

The rest of the chapter is organized as follows. Section 2 introduces the related literature review of current works. Moreover, we indicate the gaps and improvements in existing approaches. Section 3 outlines the details of the proposed framework. Section 4 gives a detailed analysis and performance evaluation. In Sect. 5, it discusses the results from the experiments. In Sect. 6, we conclude the chapter and outline the future work.

## 2 Background and Preliminaries

In this section, we discuss the limitations and challenges of the current works of security concerns in IoT and its relevant solutions.

### *2.1 The Security Concerns in IoT*

The growth of IoT brings several concerns and challenges in security and privacy areas. Because traditional systems and IoT systems have a huge difference, it is inappropriate to conduct security mitigations based on conventional strategies [13]. In a hardware layer, most sensors and actuators are limited with computational power and resources. Hence, most security mechanisms are designed to be lightweight to satisfy the resources-constrained devices [14]. In a communication layer, the protocols used in IoT systems are different from the traditional IT systems. For example, the Datagram Transport Layer Security (DTLS) is used in IoT systems, whereas Transmission Control Protocol (TCP) is used in traditional IT systems. In a service layer, data sharing is an important element to be considered in IoT systems, which may cause security problems in data privacy, data leakage and data integrity etc. The security mechanism designed for traditional IT systems is from the perspective of users. However, the security concerns in IoT systems are based on data.

Traditional security schemes cannot provide full protection for IoT devices, since IoT devices are resources constrained and easy to be compromised. Many works have been contributed to this area. In the work [15], Bertino and Islam indicated that IoT systems suffer a large number of high security risks compared with traditional computing systems. It also summarized the reasons why IoT systems are so vulnerable. In addition, it pointed out that each IoT device has at least 25 vulnerabilities on average. Similarly, D'Orazio et al. analysed that operating systems can be exploited by adversaries seeking to exfiltrate data in IoT devices [16]. In this research, it used IOS devices as a case study to highlight the security concerns in IoT systems. Both works have confirmed that IoT system is vulnerable and easy to cause security concerns. Hence, most approaches use the access control mechanism to provide a secure environment for IoT devices.

Sahraoul and Bilami proposed a host identity protocol to provide a secure network for IoT devices [17]. In this research, it reduced network overhead by eliminating unnecessary header fields. Moreover, by considering low capability IoT devices, it presented a lightweight key distribution method between IoT nodes and users. Moreover, a high resource device can on behalf of low resources devices to perform resource consuming tasks in a wireless network. The main limitation in this research is that the approach has scalability issues as high resource devices must be located within the wireless range of all IoT devices.

In the work [18], Liu et al. proposed a feasible access control mechanism for IoT systems. There are two authentication authorities in that mechanism to identify the registered users and devices, which include Registration Authority (RA) and Home Registration Authority (HRA). The IoT devices need to register with the RA for the following authentications. All users need to register with the HRA for identity authentications. If a user wants to gain data from an IoT device, it first sent a request to RA. Then, the RA interacts with HRA to check the identity of that user. If the user passes the authentication, the RA can generate a shared key between the user and the IoT device. However, in this research, it experienced scalability issues when lots of IoT devices join the network. There is a bottleneck for the RA and the HRA to store registered users and devices.

From relevant literature, it provided solid evidence regarding the security in IoT systems. First, IoT systems are extreme weakness for cyber-attacks, since most of the IoT devices are resource constrained [19]. Second, access control can provide lots of benefits for IoT systems and can present a secure environment [20]. However, considering the single point of failure and scalability issues, the centralized mechanism for access control can cause another security concern for IoT systems [21]. Hence, it is important to ensure that the security approaches must have good scalability in IoT systems.

## *2.2 The Traditional Approach: Access Control*

In this section, we present different models of the access control mechanism. By discussing each version of access control models, we analyse the security concerns in access control models and point out the feasible solutions.

### The Traditional Access Control Model

The first generation of the access control mechanism can be referred to BLP model and Biba model [22]. With the development of hardware in computers, the access control mechanism become more flexible. Some of the most representative approaches are Discretionary Access Control (DAC) [23] and Mandatory Access Control (MAC) [24]. In order to address some access issues in complicated systems, Role-Based Access Control (RBAC) was proposed to present a fine-grained access to authorized users [25]. In RBAC, it defines various roles for a user. Then, it allocates different levels of permissions based on the defined roles. It is widely used in large size of industries to meet security requirements. In Fig. 4, it presents the architecture of RBAC. With the development of IoT systems, the previously access

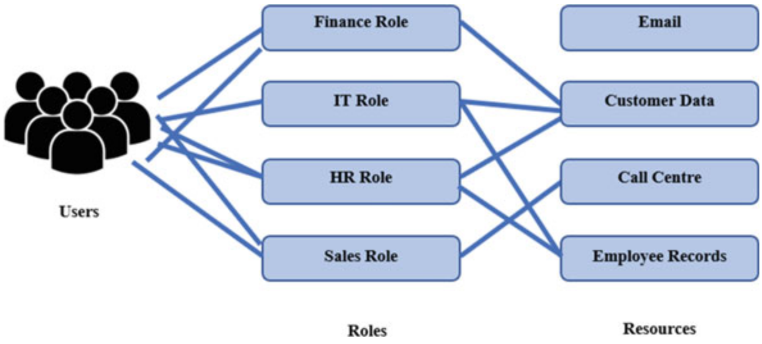


**Fig. 4** The architecture of RBAC

control model cannot solve the new challenges faced in current computing environments. The Attribute-Based Access Control (ABAC) was proposed to achieve the security requirements in IoT systems [26]. In ABAC, it grants permissions based on registered attributes of objects and subjects. In Fig. 5, it shows the general model of ABAC.

In IoT systems, ABAC model can provide more flexible and efficient management. Generally, ABAC uses a four-tuple <S, O, P, E> to represent different properties, where S is attributes of a subject, O is attributes of an object, P is attributes of permission and E is attributes of environment. A simplified ABAC architecture is given in Fig. 6. There are two stages in the ABAC, including the preparation phase and the execution phase. In the preparation phase, devices need to register their attributes into attribute authority (AA). Then, AA needs to store those attributes and create a connection between attributes and permissions. The policy administration point (PAP) uses those relationships, stored in AA, to conduct formal specifications. During the execution phase, the policy enforcement point (PEP) could receive a request from objects. It can interact with AA to check related attributes. Then, the

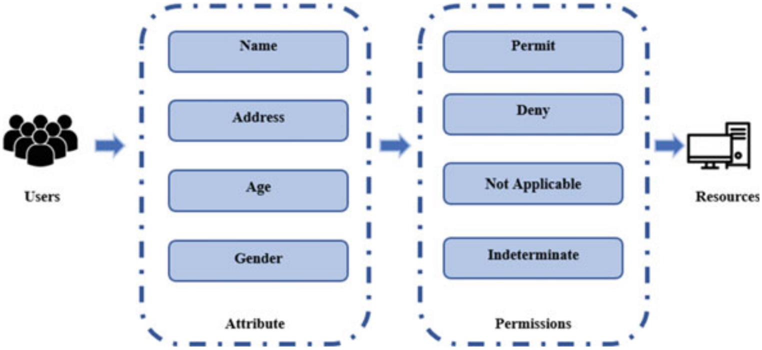


**Fig. 5** The architecture of ABAC

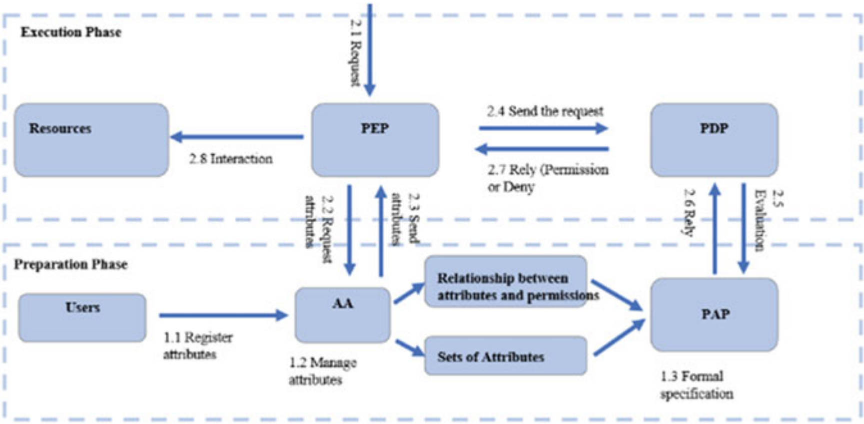


**Fig. 6** The general structure of ABAC

PEP sends information to the policy decision point (PDP). The PDP interacts with PAP to check polices. Based on the information stored in the PAP, the PEP can allocate different access permissions for the original request.

### The Limitations in Traditional Access Control Models

In order to solve the single point failure issues, some approaches [27, 28] use the distributed method to validate the right access control by requested IoT devices instead of a centralized entity. Unfortunately, IoT devices are easy to be compromised because of the low capability of memory and limited computational resources [29]. Therefore, the validated access control cannot be fully trusted. To address this challenge, blockchain technology can become a problem solver that may provide efficient trust management in an IoT environment. Moreover, by considering IoT characteristics, such as, heterogeneous network and limited computational resources [30], it is imperative to optimize the current consensus algorithm to achieve optimal performance.

## *2.3 Blockchain*

In this section, we provide a further discussion about blockchain technology. The main objective of this section is to highlight the benefits of blockchain, which can solve challenges in traditional access control models.

### Blockchain Features

By considering the fundamental structure of blockchain, we summarize key features of blockchain. Through analysing those features, we can understand why using blockchain technology is an enabler to solve challenges in the IoT environment.

*Decentralization.* In a traditional system, all data are managed by a central trusted entity. There are many disadvantages to this kind of centralization, which includes extra costs, single-point failure and bottleneck. On the contrary, blockchain can allow each user to store data locally. Moreover, it does not need a central entity to manage an entire system. The consistency of each user can be achieved by using a consensus algorithm [31]. Therefore, blockchain mitigates lots of limitations from a centralization system.

*Immutability.* There is a sequence of linked blocks to connect each block in the blockchain. Any changes in previously blocks can cause a huge modification for an entire blockchain system. At the same time, a Merkle Tree, which is a root hash, is stored in each block. It can generate a different Merkle Tree, once a tiny change happens in a transaction. Hence, it is easy to detect any falsification by comparing historical data. The hash function and the Merkle Tree can ensure data integrity.

*Nonrepudiation.* For each transaction in the blockchain, the user needs to sign the transaction by using a private key. Because the public key is publicity visible, the owner of a transaction can be verified by any user in the blockchain. Thus, the signed transaction cannot be denied by the signer.

*Transparency.* For most blockchain systems, each user has the same right to view historical data and interact with a blockchain system. The ledger of each user is publicly visible. Furthermore, each user has the right to validate new transactions and blocks, which are available for everyone. Therefore, data in the blockchain is transparent for every user.

*Pseudonymity.* Even though blockchain ensures transparency for every user, it can provide a certain level of privacy. There are lot of work focused on this area to balance a good performance between privacy and transparency. However, because blockchain addresses are traceable, it is difficult to maintain a high level of privacy for each user. Therefore, it can only use pseudonymity instead of full privacy.

*Traceability.* There is a timestamp for each transaction in the blockchain, which indicates when the transaction is generated. Since blockchain is transparent, each user can view historical data to identify the specific transaction. Moreover, users can trace the original data based on the timestamp.

### Data Structure of Blockchain

Commonly, there are 5 layers in the architecture of blockchain, which is shown in Fig. 7. In the bottom layer, it consists of different hardware, supporting blockchain technology. A layer above is the data layer, which includes various amounts of data structures to support communication in the blockchain network. Followed by the data layer, it is a network layer that is based on a peer-to-peer network. The main purpose of this layer is to ensure efficient interactions among participating nodes. The next layer is the consensus layer, which may include PoW, PoS, DPoS and other consensus algorithms. It is very important that it can make sure an agreement of

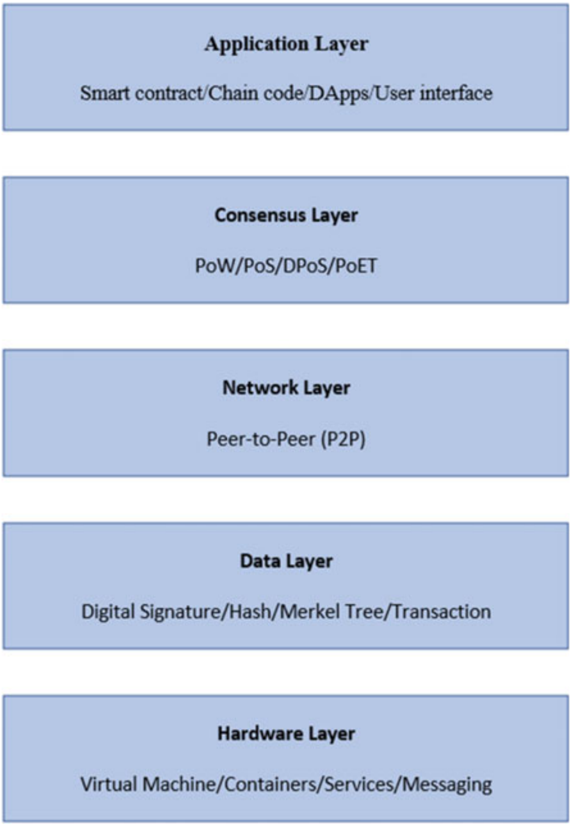

Application Layer
Smart contract/Chain code/DApps/User interface
Consensus Layer
PoW/PoS/DPoS/PoET
Network Layer
Peer-to-Peer (P2P)
Data Layer
Digital Signature/Hash/Merkel Tree/Transaction
Hardware Layer
Virtual Machine/Containers/Services/Messaging

**Fig. 7** The structure of blockchain

states among different nodes in the system. The top layer is an application layer that may include smart contracts, DApps and other APIs.

In a blockchain network, each block has the hash of all information of its previous block, as illustrated Fig. 8. Also, each node maintains a copy of the global state and

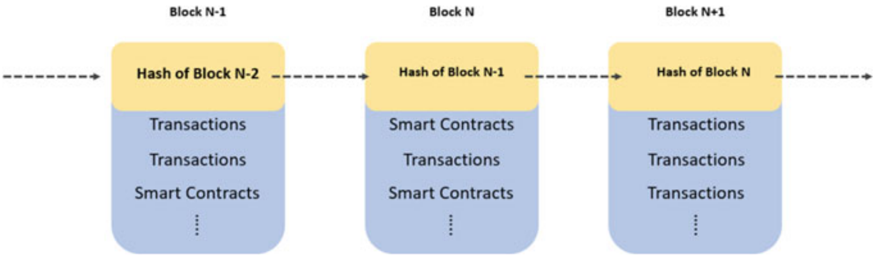


**Fig. 8** The structure of blocks

a copy of the ledger. The nodes, which generate new blocks, are called miners. In Bitcoin and Ethereum, each miner creates a block and executes the consensus algorithms. Then, the miner broadcasts its generated block to other nodes for validation. After being successfully verified by other nodes, that block can be formally added to the blockchain. To protect the participating nodes in the blockchain, IBM introduced a Hyperledger Fabric platform, where only the authorized nodes can form the blockchain [32]. Zhang et al. proposed a framework which all access control policies and IoT devices are managed by smart contracts [33].

There are mainly two parts in a block, which contains a block header and a block body, as shown in the Fig. 9. In the block header, it can record each transaction by using hash technology. The ultimately hash value of the transactions is stored in the Merkle root. Besides that, it contains a hash value of the current block, which is used for integrity checks. The pre-hash is the hash value of the previous block. Since each block may be created by different users, the pre-hash can be used to ensure no one change any data in that block. There is also a time scheme, which is used as an identifier to locate each block. Since the blockchain system is publicly visible, every participating node can view the data at any block. The connection between the block header and the block body is called the Merkle Root. In the bottom part of Merkle

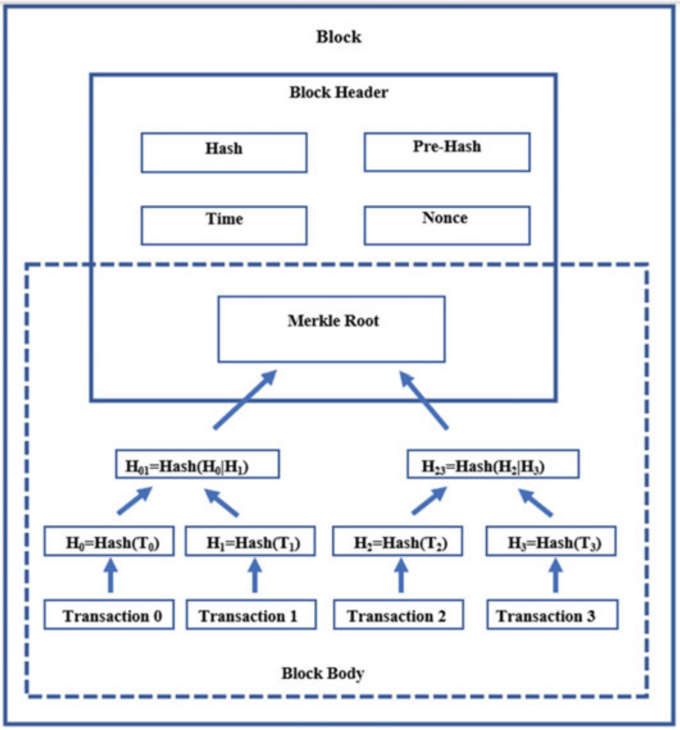


**Fig. 9** The elements in a block

Tree, there are recorded transactions in a block. A level above, it is labelled with the hash value of its child node. In the next level, every two hash values need to be hashed again to generate another hash value. In this case, the root of the Merkel Tree is used to provide a secure verification of the contents of the recorded transactions. Once a transaction is changed, the root also needs to be changed.

### Consensus Mechanism

Unlike traditional systems, blockchain systems do not rely on a third party to construct trust among each participating node. In a blockchain system, the consensus algorithm can ensure reliability and consistency among all nodes. In blockchain systems, there are mainly four major consensus algorithms, which include Proof of Work (PoW), Proof-of Stake (PoS), Delegate Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT). There are also other consensus algorithms, which may include Proof of Trust (PoT), Proof of Elapsed Time (PoET) and so on. However, those consensus algorithms have not been widely used in industries. At the two most popular platforms, Bitcoin and Ethereum, PoW is used as the consensus algorithm.

In Bitcoin, each node needs to consume computational resources to calculate the solution of a puzzle. The puzzle is usually a computationally hard but easy to be verified problem. Hence, by solving the puzzle, the node can become the miner and have the right to broadcast its block. From the Fig. 10, each node in Bitcoin can easily verify the correctness of the solution provided by the miner. After successful verification, the block can be appended to the blockchain. There are some limitations in PoW, which include consuming too many computational resources and low throughput. In order to solve those challenges, PoS was developed in Ethereum with the purpose to instead of PoW in the future. Contrary to PoW, PoS requires the miner to pay a certain amount of cryptocurrency to prove the credibility of the data. Essentially, the more assets the miner has, the easier the puzzle that needs to be solved. In this case, the miner does not need to consume too many computational resources. Thus, the throughput of the entire blockchain system also can be increased.

The PoS can achieve excellent performance in Ethereum. However, the selection of a miner is limited to a small group of nodes, which means the blockchains system

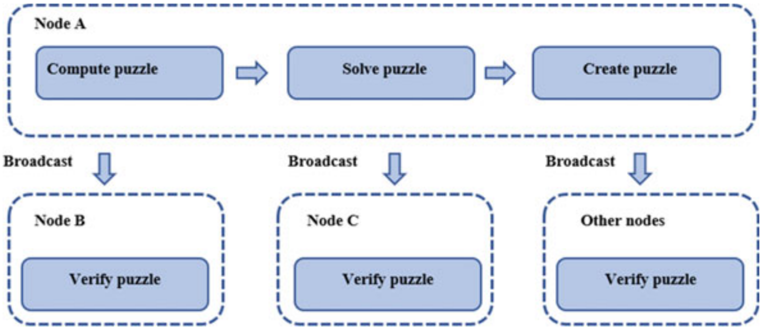


**Fig. 10** The consensus mechanism in Bitcoin

suffers a certain level of centralization. Then, the DPoS was proposed by using the voting and election process to prevent blockchain from centralization and malicious nodes. In DPoS, each node votes for delegates based on their weights of assets. The selected delegates can manage the entire blockchain system. In this case, the speed for creating and validating blocks is much faster. Thereby, the throughput can be highly improved with the DPoS.

Unlike PoW, PoS and DPoS, PBFT is widely used in consortium blockchain. The purpose of PBFT is to solve the Byzantine generals' problem in a distributed network. There are three phases in PBFT, which include Pre-prepare, Prepare and Commit. In the Pre-prepare, the leader sends the request to all nodes once it receives the request from the requester. Then, in the Prepare stage, each node broadcasts the request to each other again. Finally, in the Commit stage, the value can be committed as more than two-thirds of nodes agree. Obviously, PBFT has limitations in scalability, because each node needs to interact with each other for consensus. However, it has a low possibility to cause a fork.

### Communication Models

Basically, there are two steps for the mining process, Firstly, a miner needs to calculate a computational puzzle and find a valid hash value for the block. Then the hash value is stored in the miner's local transaction pool and broadcasted the solution to the puzzle to the entire network. Secondly, the other nodes quickly check the validity of the broadcasted solution. Once the solution is correct, the miner can add its own block to the blockchain. The following graph shows the validation process between the two parties in Fig. 11. During the calculation process, the value of nonce is used to determine the difficulty level of the current puzzle. The length of the blockchain keeps increasing as the mining process is conducted in the network.

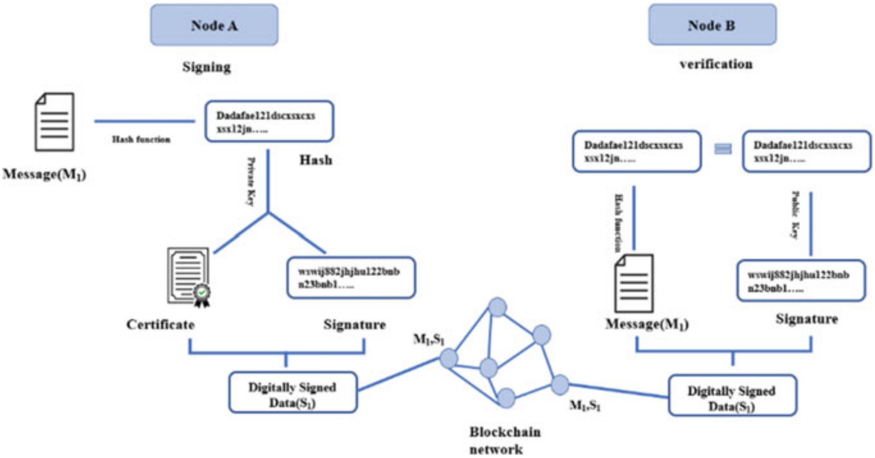

Node A
Signing
Message($M_1$)
Hash function
Dadafae121dscxsxcxs xsx12jn.....
Hash
Private Key
Certificate
wswij882jhjhu122bnb n23bnb1.....
Signature
Digitally Signed Data($S_1$)
$M_1$,$S_1$
Blockchain network
$M_1$,$S_1$
Node B
verification
Dadafae121dscxsxcxs xsx12jn.....
Dadafae121dscxsxcxs xsx12jn.....
Hash function
Public Key
Message($M_1$)
wswij882jhjhu122bnb n23bnb1.....
Signature
Digitally Signed Data($S_1$)

**Fig. 11** The validation process between two nodes

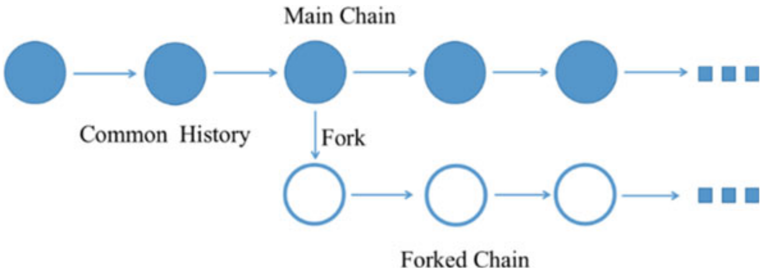


**Fig. 12** Blockchain forking

During the mining process, it is easy to generate a sub-chain in the blockchain system, which is called a fork. It is a serious security problem in the blockchain, which attract much attention from researchers and specialists. Since the blockchain is a distributed system, it has a high possibility that two valid solutions are found at the same time. Both of those situations can cause a fork in the blockchain. As shown in the Fig. 12, the main can have multiple sub-chain at the same time. At this stage, miners are free to choose any chain to append their blocks from their perspective. With the nature of consensus algorithms, there is only one longest chain that exists in the network. In this case, all the miners will append their blocks on the top of the longest chain.

The security of the blockchain is based on the assumption that most of the participating nodes are honest. In other words, most computational resources are controlled by honest nodes. For example, in Bitcoin, there is an incentive mechanism that encourages participating nodes to be honest. The driving factor for the miners is the reward (12.5 Bitcoins) that they can receive as a block can be appended to the blockchain. Since miners need to solve a crypto puzzle, the probability of solving a puzzle is proportional to the number of computational resources used. Hence, some individual miners can join together to accumulate their computational resources, which are called mining pools. Once a mining pool successful mine the blocks, all associated miners can share the reward based on the contributed computational resources.

## *2.4 Blockchain Integration with Access Control*

There are several works have proposed blockchain-based access control, which aims to remove the centralized entity and avoid a single point of failure. In the work [34], Dorri et al. proposed an access control mechanism with blockchain technology, in which each home miner has a private blockchain with a policy header string access control policies to control all the access requests related to the home. However, the computing ability of blockchain was not fully used in this scheme and was largely wasted. Moreover, Maesa et al. proposed an approach to manage access control in a distributed way based on blockchain technology [35]. The limitation of that approach is that only uses blockchain as a database to store the access control policies. The computing capability of blockchain has been fully used in the work regarding access control [36]. Ramachandran and Kantarcioglu considered blockchain as a platform to provide data management and verification. They recorded immutable data by using smart contracts and an open provenance model (OPM). Recently, Zhang et. al proposed a smart contract-based framework to manage all the access control through a blockchain network [33]. By implementing several smart contracts, it can achieve distributed and trustworthy access control schemes in the IoT environment. However, the system did not provide the direct interaction between IoT nodes and the cost to deploy smart contracts is relatively high.

## 2.5 Blockchain Integration with Trust Management

It is necessary to introduce a trust evaluation mechanism into the blockchain-based network. Trust management between peers and services can mitigate possible security attacks. Trust relationships are built by evaluating the honesty of peers and services based on their behaviours and the inter-node interactions. The nodes can receive low trust scores and corresponding penalties when they provide malfunctioning services or violate predefined rules.

Since IoT devices can join and leave the network at any time, it can cause a burden to manage all the data flows from each device. Therefore, it is quite challenged to manage all trust scores in an IoT environment. However, one of the benefits of using blockchain is that it can ensure data integrity and non-repudiation. Once data is stored in a blockchain network, it cannot be modified or removed according to the characteristics of the blockchain. Thus, trust scores can be stored in a blockchain network to optimize the storage issues. From the work [37], Shala et al. introduced a secure approach to store the rated trust scores in the blockchain network. Moreover, it solves several security problems in the trust evaluation process and blockchain network. It provides some inspiration regarding data storage.

Data stored in a blockchain network can be used for integrity check-ups. In the work [38], Steinheimer et al. presented a P2P overlay network to store the current status of trust scores for each node. Moreover, each node has permission to write the data into the system. At the same time, the node can check the history of trust scores from a P2P overlay network and compare those with data derived from the blockchain. Thus, the blockchain can ensure the immutability of the stored data.

## 2.6 Trust-Based Consensus Algorithm

Currently, there are many literature working on the trust-based consensus algorithm. In the work [39], Zou et al. combined accountability mechanisms with online services by implementing the blockchain concept to ensure the immutable and traceable trust management and origin of data. In their approach, they used a hybrid blockchain system, which includes a consortium blockchain and a permissionless blockchain. The consortium chain consists of operators, regulators and other stakeholders. It is used to manage the whole system. The permissionless chain is open to the public, which is used for dynamically validating transactions.

In the work [40], Bahri et al. proposed another approach to a trust-based consensus algorithm. In this approach, the trust scores of each node can be derived from the trust graph which is created from each node rates trust scores to other nodes. The higher the trust score for the participating node, the more energy can be waived to use the PoW algorithm.

Shala et al. tried to modify the above-mentioned limitations in this section and proposed a consensus mechanism in a decentralized community, called Trust-CP [41]. The main benefit of this consensus algorithm is that it is based on a dynamic trust model to evaluate the trust scores of participating nodes in the network. Moreover, blocks and transactions can inherit the same trust scores from the sending nodes.

There are many reputation-based consensus algorithms used in different industries. However, most of them are suffered from low efficiency and high consumption. In the work [42], Watanabe et al. proposed a new consensus algorithm based on the PoS. The algorithm is based on the collapse of credibility. The node that has the highest credit score can generate and validate blocks and transactions. Similarly, in the research [43], Gai et al. proposed a reputation-based consensus algorithm, called Proof of Reputation (PoR). In this mechanism, it indicated that the node, that write transactions into a block, is based on the reputation score. Furthermore, the reputation score can be regarded as the motivation mechanism, as most of participating nodes want to write the block into a blockchain. Wang et al. proposed a consensus algorithm which is not only based on the reputation scores, but also considers encouraging nodes to perform in good behaviours [44]. Moreover, this mechanism inherits the most properties of PoW, except changing the coin incentive to a reputation incentive. However, the authors claimed that this algorithm can cause a reduction in users of the public blockchain system. In the work [45], Yu et al. presented a consensus algorithm, called RepuCoin. It is based on a weighted voting algorithm, in which the weight of a vote is the percentage of that node's reputation score.

## *2.7 Security Attacks in Blockchain*

Basically, there are four different cyber-attacks related to blockchain. They include ledger-based attacks, peer-to-peer network attacks, smart contract-based attacks and wallet-based attacks. In the ledger-based attacks, a fork can cause extremely detrimental damage to a blockchain network. In order to prevent such an event happened in the blockchain, the system needed to be fully decentralized so that there is no single participating node can take control of the entire network. The ledger-based attack includes the 51% attack[46], double spend attack [47] and Finney attack[48].

One of the basic components of blockchain technology is the peer-to-peer network. The purpose of a peer-to-peer network is to ensure that each node can hold a copy of the ledger of transactions. After a consensus period, it can ensure that the ledger of each node can achieve identical on the peer-to-peer network. Hence, a peer-to-peer network plays a vital role in the blockchain network. There are three attacks related to the peer-to-peer network attack, which include the Sybil attack [49], the Eclipse attack [50] and the Distributed Denial-of-Service attack (DDoS) [51]. If there is a fault in a smart contract, it can cause damage to the related transactions. Furthermore, it may affect millions of currency from participating nodes. The most famous attack, called the DAO attack, cause 70 million US dollars lost by conducting a recursive withdraw function [52]. In the Ethereum platform, each node has its own wallet to pay the transaction fees. To obtain wallet credentials, attackers can attack the node's wallet on the blockchain.

## 3 Blockchain-Based Trust Management Framework

In this section, we discuss our blockchain-based trust management framework in detail. Firstly, we discuss the overview of the framework, which is presented in Sect. 3.1. Then, we illustrate the structure of trust management used in our approach as outlined in Sect. 3.2. Following that the architecture of transactions and blocks is presented in Sect. 3.3. Next, we discuss the reputation-based consensus algorithm in Sect. 3.4. The reputation-based verification mechanism is presented in Sect. 3.5. In Sect. 3.6, we give a summary of our architecture.

### *3.1 Overview*

We propose several smart contracts to evaluate the behaviours of each participating node in our framework (to meet requirement R1). Besides that, current consensus algorithms are too complex. In order to meet the requirement (R2), we propose a reputation-based consensus algorithm and only include IoT gateway devices in our framework. The other nodes, which we call IoT devices, can connect with their own IoT gateway device for local communications. As shown in Fig. 13, the framework is a topology of a blockchain-based IoT network. Furthermore, there is a significant delay associated with ensuring that a transaction is confirmed by nodes participating in the blockchain. In addition, there is a time requirement for IoT devices. We propose a reputation-based verification algorithm that decreases the delay time in our framework (to meet requirement R3). In this chapter, we assume IoT gateway nodes have good performance and high resources. Blockchain communication only occurs among all IoT gateway nodes, since IoT gateway nodes can support computational capabilities in blockchain systems. Other nodes that are connected with IoT gateway nodes are called IoT devices.

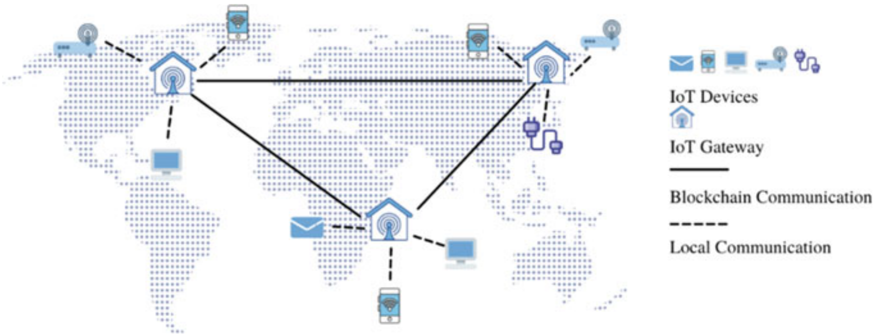


**Fig. 13** The blockchain-based IoT network

In order to reduce packet overhead in transactions, we separate the transaction flow and the data packet flow. In our approach. the first step is to find the corresponding IoT gateway node by the transaction flow. In this process, there is no extra data that need to be filled into the transaction flow. In the next step, the corresponding IoT gateway node can use a routing protocol to reach the requester node. The data flow uses the optimal route by using a network routing protocol, such as, OSPF. To ensure the paths are unicast, the identifications of each IoT gateway node are known in the blockchain. In this case, separation of the transaction flow and the data flow can reduce the delay time in the IoT environment.

## *3.2 Smart Contract Systems*

The requirement R1 is satisfied by incorporating smart contacts. The detailed information is explained in the following sections. The main purpose of smart contracts is to manage a trust environment in our framework. As mentioned before, ABAC is one of the most efficient mechanisms used in IoT systems. Moreover, the attributes of each IoT gateway node can be used as identifications. Hence, we incorporate ABAC with blockchain technology in our framework. As shown in Fig. 14, there are 5 different smart contracts used to manage the trust evaluation system. The Management Contract (MC) provides functions to manage all the contracts. The Attribute Contract (AC) stores all the attributes of each IoT gateway node. The Access Control Contract (ACC) implements access control policies for a pair of nodes. The Reputation and Trust Contract (TRC) is to evaluate the behaviours of each IoT gateway node. The Judgement contract (JC) interacts with the ACC to judge misbehaviours and determine corresponding penalties. Our framework only consists of one of each smart contract. Consequently, the reputation score can dynamically reflect the behaviours of each IoT gateway node.

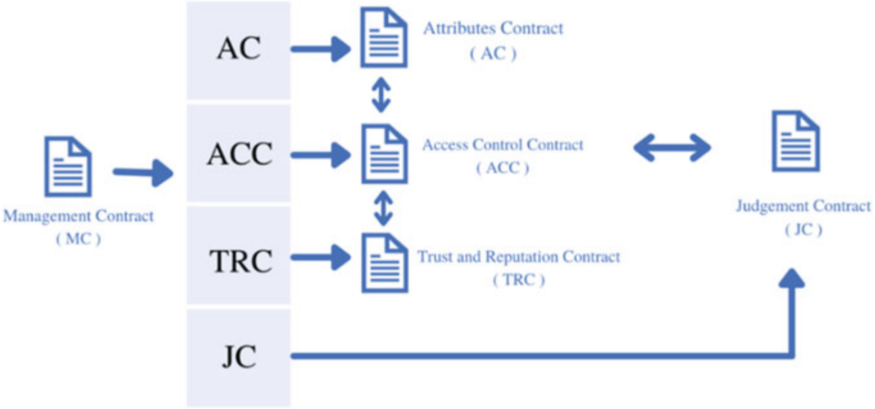


**Fig. 14** The architecture of our blockchain-based system

### 3.3 *Transactions and Blocks*

In this section, we introduce our data structures, which also can provide a secure environment for all IoT gateway nodes.

The transactions in our framework record all the situations when one IoT device wants to access another IoT device. The transactions are secured by asymmetric encryption, digital signatures and cryptographic hash. Traditionally, each transaction only needs to have a single signature from the IoT node to demonstrate its legitimacy and validity. We also consider another circumstance, which needs multiple signatures from a group of nodes. The multisig approach can enhance the security of transactions. In the multisig model, there is required at least n signatures out of from m IoT nodes. The more signatures involved, the more secure of the transactions. Moreover, multisig can help the node, that lost its private key, to retrieve the private key. Hence, in our framework, we consider both single signature and multisig in our structure of the transaction.

The structure of a transaction is shown as follow:

Con_ID || Tx_ID || Prev_ID || Gen_Hash ||Timestamp|| Data || Sign

Where Con_ID is the identification of the smart contract. The MC directs the request to the right smart contract. For example, if a node wants to update its attributes, it needs to contact the MC. All the requests go to the MC first. Then, based on the identification, the MC can direct the updated request to the AC. Tx_ID is the hash of the content of the current transaction. It is a key element in the cryptographic link. Moreover, it can be used as the unique identifier of the current transaction. Prev_ID is the hash of the content from the previous transaction. It can ensure that all the transactions are linked and cannot be modified once added to the blockchain. Gen_Hash is the hash of the generator's public key, which can be used for later verification. The Timestamp is the time when the transaction is created. Data is the detailed information related to access control policies. For example, if a node wants to access a file in another node. The whole process can be recorded in the data field in the transaction. The Sign is the signature of the subject. It can ensure that this transaction is initiated from the IoT node. For multisig, there are at least n out of m nodes' signatures in the Sign field.

When the number of transactions achieves the maximum number that a block can store, those transactions can form a block and be added to the blockchain. All the transactions can build a Merkle Tree and the Merkle Root is stored in the header of the blockchain. The structure of the header in the blockchain is shown as follow:

B_Hash || P_B_Hash || Merkle_Root ||Timestamp|| B_Miner || B_Validators

B_Hash is the hash of the content of the current block. P_B_Hash is the hash of the content of the previous block. Merkle root is the hash of all hashes from all the transactions. In each block, the Merkle root can provide integrity. Once a change happens in the past transactions, the Merkle root can be changed. In this case, it can also protect a blockchain from modification. The Timestamp is the time of the creation of the current block. B_Miner indicates the signature of the gateway node which generates the current block. Lastly, B_Validators presents the signatures of nodes involved in the verification.

### *3.4 Reputation-Based Consensus Algorithm*

In this section, we propose a trust-based consensus algorithm to meet the requirements in the IoT environment.

In our framework, we aim to encourage each node to be honest. The higher the reputation score of the node, the larger possibility for the node can become a miner. In this case, we can ensure that transactions in the ledger from the miner have a very high possibility to be real. Therefore, the first part of our consensus algorithm is that the potential miners can be determined from a reputation score threshold, which is defined by the designer of the blockchain. It means that the reputation score of the nodes, which needs to be higher than the reputation score of the threshold, has the possibility to become a miner. According to this rule, there are several gateway nodes that have the chance to become the miner, which we call those nodes as potential miners. At the same time, we need to ensure the miner is selected randomly from potential miners. More importantly, the number of blocks should be limited from each node. This mechanism can ensure that malicious nodes cannot continuously generate fake blocks. Therefore, the second part of our consensus algorithm is that each potential miner has a sleep-time, which is similar to a lottery system, prior to generating a new block. In the sleep-time mechanism, each node is allocated a random sleep time. Regarding the first part of the consensus algorithm, the potential miners, who awake first, can become the miner of this round. In general, our proposed reputation-based consensus algorithm ensures that all the nodes are motivated to become honest and have a higher reputation score. Furthermore, we introduce randomness during the selection of the miner.

Once a miner generates a block, it broadcasts the new block to the entire network. The transactions in each gateway node may match some or all transactions in the generated block. Each node can remove the identical transactions from its transaction pools, because those same transactions are stored in the miner's ledger. In our framework, it may be possible that several potential miners wake up at the same time and generate multiple sub-chains and cause the blockchain forking. We use the same strategy as Bitcoin called the longest ledger. The chain with the longest blocks can be regarded as the main blockchain. This will prevent the blockchain from forking.

In our consensus algorithm, we need to ensure that there is a low possibility to generate forks. Hence, we need to strictly limit that only one block can be generated during a consensus period, which will be addressed in the next phase. Moreover, the minimum value for the consensus period must be larger than the maximum end-to-end delay time in the blockchain network. Otherwise, it has a very high possibility to cause a fork if the consensus period is less than the maximum end-to-end delay time in the blockchain network. The maximum value for the consensus-period set to be P minutes, which needs to be suitable for the time requirement in the IoT environment.

### *3.5 Reputation-Based Verification Mechanism*

From the previous sections, we understand that the entire verification process may be time consuming. IoT devices can suffer a time delay. We need to propose an optimal approach for a verification scheme used in the IoT environment. In this section, we present a modified version of verification that can overcome some limitations in traditional blockchain systems. Those limitations may contain time and resource consumption. For example, it may take up to 30 min for a transaction to be confirmed in the blockchain.

Recall that we use the Burn Coins approach to demonstrate the validity of the public key. It is convenient for each node to check the legitimacy of the provided public key. The genesis transaction generated from each node contains information on the public key. In the meantime, it is compulsory for each node to conduct a genesis transaction as it joins the blockchain network. The burn coins approach is included in the genesis transaction. Hence, each node knows the public key from others.

The verification contains two main parts. The first step is to verify the validity of the block. Next, we need to verify the transactions included in that block. Each node needs to verify the newly generated block once it receives from other nodes. The initial thing that needs to be validated is the signature of the block generator. Since each node knows the public key of each other in our framework, the signature of the block generator can be verified by checking the corresponding public key. The

**Table 1** The validation mechanism in our architecture

| Reputation score | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 | 4.5 | 5 |
|---|---|---|---|---|---|---|---|---|---|
| Needed to be validated (%) | 100 | 90 | 80 | 70 | 60 | 50 | 40 | 30 | 20 |

following step is to validate the transactions in that block. We consider the block is valid only if the transactions contained in the block are valid.

During the verification process in transactions, the first step is to verify the hash value of the public key from the transaction generator. In our data structure, the hash of the transaction generator's public key is recorded in Gen_Hash. In the meantime, the public key of each node can be assessed by any node in the blockchain. The public key of the current transaction can be compared with the Gen_Hash from previous transactions in the same ledger. The validity of the public key can be demonstrated. Following by that, we need to verify the signature of the current transaction by using the public key.

In an IoT environment, the number of gateway nodes is expected to be large. Hence, many resources and time are required to verify all the transactions and blocks. It can also cause a very seriously scalability problem in the blockchain network. The overall performance of blockchain may drop significantly. To address this challenge, we propose our reputation-based verification scheme. We randomly verify a certain percentage of newly generated transactions in a block based on the reputation score of the miner.

In our approach, all the nodes need to be involved with the verification process. We also assume that with a higher reputation score of the miner, the IoT gateway node is more honest and trustworthy. In this case, the miner has a very low possibility to generate false transactions in the blocks. As shown in Table 1, it presents detailed information regarding how many percentages of transactions needed to be verified in the newly generated block. In this structure, we assume that 1 is the lowest reputation score that a node can get, whereas 5 is the highest reputation score that a node can get. Each IoT gateway node begins with a 5 score when it joins to the network. We assume that each IoT gateway node is honest. All the transactions needed to be verified for the node, which gets the lowest reputation score. Even if a node can have the highest reputation score, there is still 20% of transactions needed to be verified. We still need to consider unpredictable situations with high reputation scores of IoT gateway nodes. Therefore, it can enhance the security of our proposed blockchain-based framework.

### *3.6 Summary*

In this section, we summarize all the aforementioned features included in our proposed framework. Firstly, an IoT gateway node can either receive a transaction or a block from other nodes. If an IoT gateway node receives a transaction, it first verifies the transaction by checking the valid signature from the sending node. Second, if it is a valid signature, the node can store that transaction in its own transaction pool, which is the place of pending transactions. If it is not a valid signature, the transaction can be dropped. Third, when the number of transactions exceeds the T_max in the transaction pool, those transactions can form a block. The consensus algorithm can be triggered once a block is formed from an IoT gateway node. Forth, a miner is to be selected based on our reputation-based consensus algorithm. Lastly, our reputation-based verification mechanism is to be implemented among nodes to verify the newly generated block. The block can be appended to the blockchain once the block is valid.

In the verification scheme, we trade off the security in the blockchain system to reduce the delay time in the IoT environment. In this part, we present an example to further explain the consensus algorithm and the verification process. As shown in the Table 2, there are 5 IoT gateway nodes in our framework. We assume that each IoT gateway node has its own reputation score at the current time t. Next, there is a threshold value of the reputation score, which is the first step of the consensus algorithm. We do not focus on how to determine the value of a threshold score in this chapter. The IoT gateway, which is higher than the threshold score, can become the potential miner. In the second step of the consensus algorithm, each IoT gateway has a sleep time. From the graph below, there is a randomized sleep time to allocate to each IoT gateway node. The first awakened IoT gateway node will be the miner in this round. Hence, the IoT gateway node A is the miner of this time period. Next, more than half of the members need to validate the legitimacy of the mined block in our framework. According to the verification scheme, for the miner with a reputation score of 3.5, there are 50% of transactions needed to be verified in a block.

**Table 2** An example of the consensus algorithm and verification scheme

| | IoT gateway A | IoT gateway B | IoT gateway C | IoT gateway D | IoT gateway E |
|---|---|---|---|---|---|
| Reputation score | 3.5 | 4 | 2 | 2.5 | 4 |
| Threshold score | 3.5 | | | | |
| Potential miner | Yes | Yes | No | No | Yes |
| Sleep time | 3s | 4s | 4s | 8s | 7s |
| Miner | Yes | No | No | No | No |
| Validation scheme | Verify 50% of transaction in a block | | | | |

# 4 Framework Analysis

We analyse our framework based on the properties of validity, liveness, scalability, fairness, and security in general.

## *4.1 Validity*

In the blockchain system, it is important to demonstrate the legitimacy of the public key. In our framework, we use “Burn Coins” to authenticate public keys from each IoT gateway node. Since only legitimate users would like to burn coins to demonstrate the validity of their public keys, attackers are unlikely to take financial risks. In order to verify transactions, each IoT gateway node can compare the public key from the burn coins transaction with the public key of the received transactions. Following by that, each IoT gateway node can also verify the signature of each IoT gateway node.

## *4.2 Agreement*

Agreement means that all members in the blockchain system need to agree on the same set of transactions during a consensus period. It means that the blockchain will not permanently fork. In a consensus period, it involves two stages, including the miner selection and the block verification. The miner is selected based on the reputation score, which demonstrates the miner is honest. Since the miner is honest, the generated block has high likely to be verified as real. By considering a specific miner can be compromised by attackers, there is a sleeping mechanism for each IoT gateway in the consensus algorithm. It can ensure that different miners can be selected at each consensus period. The consensus algorithm can maximumly avoid the collision attacks, since the miner is always changing in our framework. In the block verification, we assume that at least half of the IoT gateway nodes are honest. In an IoT environment, time is a criterion that ensures all the services can be satisfied. The worst-case can be considered that a team of attackers prevent a block from being verified. It can cause extremely time delay and affect the performance of our framework. However, there is at least half of the IoT gateway can verify the block. In this case, the block can be effectivity appended to the blockchain. Therefore, our framework can always guarantee the agreement property.

## 4.3 Liveness

Liveness means that our framework can show the latest information on the properties of IoT gateway nodes. There are two parts related to liveness in our framework. One is the reputation and trust evaluation for each IoT gateway device, and another is about the consensus algorithm. In the reputation and trust evaluation phase, we need to always update the reputation score of each node. Based on the formulas in Sect. 3.2, the value of a reputation score of each IoT gateway node always changes within a period. The updated value can directly reflect the honesty of each IoT gateway node in the blockchain. Recall that we assume that the higher value of the reputation score, the more honesty of the IoT gateway node. Therefore, the reputation score is dynamically changed in our framework. In a consensus period, there are two steps to select the miner. The first step is to select the potential miner, which is based on the reputation score of each node. Since those reputation scores keep changing, the range of potential miners is different from each consensus period. In the second step, we have a sleeping mechanism that each IoT gateway node randomly awakes during a consensus period. In other words, the qualified node is selected to be the miner based on a truly random system. In this case, the miner cannot be the same IoT gateway node at an adjacent time slot. This prevents the miner from storing transactions locally. Furthermore, it can cause a fork in the blockchain by storing lots of transactions locally. There is an extreme case where two nodes can be selected to be the miner simultaneously at the second stage. This is a very rare event that can happen in our framework, because the sleeping time allocated to each node can be precise to a microsecond. Therefore, the second stage of the consensus period can achieve liveness. Consequently, our framework can always guarantee the liveness property.

## 4.4 Scalability

Since the blockchain network needs to incorporate lots of IoT devices, scalability is one of the main issues to be considered in our framework. good scalability means that the performance of a blockchain network has slight influence on the number of IoT devices. In our framework, we only use IoT gateway devices as the nodes to establish the blockchain system. In this case, the involved devices are reduced. Besides that, in order to reduce packet overhead, we separate the transaction flow and the data flow. Hence, there are not lot of data involved in each transaction.

## 4.5 Fairness

Fairness means that each IoT gateway works normally and behaves honestly. Any abnormally and malicious activities are considered unfairness. The fairness of our framework can be evaluated from several aspects. The first part is to check if the evaluation of reputation and trust scores is trustworthy. The second part is to check whether the consensus process is neutral and impartial. The third part is to see if our framework can defend against collision attacks.

## 4.6 Security

In this section, we discuss the security features of our framework, Firstly, we identify the capability of the threat model. With the threat model, we discuss the possible security attacks to which IoT networks are particularly vulnerable. Lastly, we point out how to defend against those attacks by using our framework.

### Threat Model

The adversary can be any IoT gateway node or IoT device in the blockchain network. The capability of adversaries includes data capture, discarding transactions, generating false transactions in the blocks, generating fake trust and reputation scores, creating fake attributes, deanonymizing IoT devices, creating fake sleeping time, and signing fake transactions. The adversaries can be a group of IoT gateway nodes and IoT devices. They can consume the resources of participating nodes by flooding the network. More importantly, we assume that the encryption schemes cannot be compromised in our framework, which means that the public-key cryptosystem is assumed secure.

### DDoS Attack

In a DDoS attack, the attacker normally sends a large number of requests to IoT gateway devices and IoT devices to consume resources. The target IoT devices cannot interact with other devices for genuine transactions. In this case, our framework has a unique way to defend against this attack. Recall that there is a smart contract called ACC. The main duty of this smart contract is to monitor the behaviours of nodes in our framework. Once it finds unexpected requests, it would record information of the requesting and send this report to JC. The JC can implement punishment to the malicious nodes. By monitoring behaviours, our framework can defend against DDoS attacks in different scenarios.

### Sybil Attack

In our framework, it is compulsory that each IoT gateway device needs use "Burn Coins" to demonstrate the legitimacy of the public key. Even though each IoT gateway node has multiple accounts, those accounts are derived from a pair of public and private keys. In this way, validators can verify the authenticity of the public key of each transaction. By checking the public key, the validators can later verify the signature of each transaction. Hence, fake transactions can be detected easily. Moreover, the reputation score and the sleeping time are hard to be compromised. This means that attackers need a large effort to become the miner and broadcast the fake block. Consequently, a Sybil attack is hard to be conducted in our framework.

## 5 Experiment

Based on Sect. 4, we provide a theoretical analysis of evaluation for our framework. From the analysis of the validity, liveness, scalability, fairness and security, our framework can theoretically meet the requirements. In this section, we will take experiments to demonstrate that our framework meets the previously three requirements, which include the behaviours of IoT gateway nodes that need to be evaluated, the consensus algorithm does not consume lots of resources and the time in a consensus period needs to be reduced. Therefore, we conduct simulated experiments to evaluate security in our framework.

The device that we used in our experiment is MacBook Air with CPU is Intel Core i5, 1.6 GHz and Memory is 16 GB. The programming language is used Python to implement the clients and the server applications. There are two main aspects that are simulated in this chapter. The first part is to establish the basic communication network for all IoT gateway nodes. We need to generate several Clients and one Server connected using TCP Sockets. The clients represent IoT gateway nodes. The Server acts as a broker to pass all the messages among all IoT gateway nodes. The Ip address of the Server is 192.168.1.1. The clients can connect with our framework to receive messages from the blockchain. Each IoT gateway is allocated a different IP address, from 192,168,1.2 to 192.168.1.9. The second part is to construct our framework by using the algorithms proposed in Sect. 3. In this step, we use Python to construct the consensus algorithms and verification scheme. Smart contracts are created by using Solidity. The topology of the network is shown in Fig. 15.

Recall that the range of reputation scores is from 1 to 5, with 1 as the lowest grade and 5 as the highest grade. The threshold in our consensus algorithm is 3.5, which is used to find relativity honest IoT gateway nodes. Since we only need to demonstrate the feasibility of our framework in an IoT environment, only a few IoT devices are needed to demonstrate if the requirements (R1–R3) have been met in our experiment. Thus, we provide a small-scale system to demonstrate that our framework meets the previously mentioned requirements. The IoT gateway devices can take up to 8 devices in this chapter. The above settings are considered as default configurations in our experiment. Unless explicitly specified, the above-mentioned settings are used in all simulations.

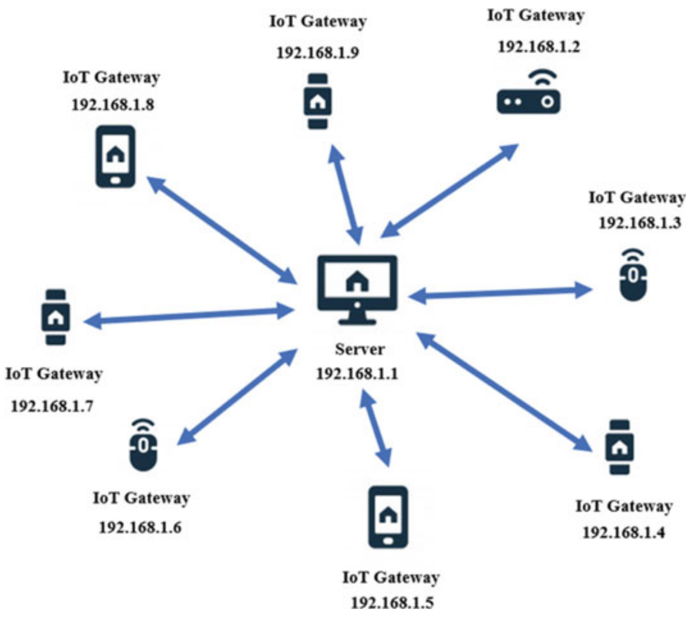


**Fig. 15** The topology of the network in experiments

## *5.1 Delay Time*

We measure the delay time in our framework with an IoT gateway device performed with multiple transactions. The time is measured from the request generated to the response received. We assume that each IoT gateway node can only interact with its neighbours. In this way, the simulated network can perform a P2P communication. The parameters are the time from the requester sending a request to the receiver's response for that request. In this experiment, we only need to simulate the data packet that is transmitted among IoT gateway nodes in the blockchain network. The routing protocol used in this experiment is OSPF. We assume that there are no malicious activities in the experiment.

We first need to establish a baseline for the experiment. Recall that we separate the transaction flow and the data flow in our framework. Therefore, the baseline is the approach that combines the transaction flow and the data flow. This approach is widely used in many traditional blockchain systems, such as, PoW and PoS. The main reason is that there is not much data to have interacted with each other nodes in traditional blockchain systems. In this chapter, we implement blockchain technology in an IoT environment. Since there are lots of sensors and actuators to collect and transmit data, we need to pay more attention to the efficiency of our framework. The transaction flow does not carry so much data in its data structure. The purpose of the transaction flow is to find the target IoT gateway device. After the target IoT gateway

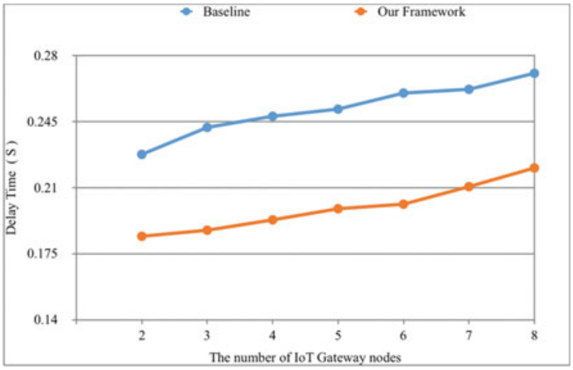


**Fig. 16** The delay time in our framework

device has been found, those two gateway nodes can interact with each other by using OSPF.

We compare our framework with a baseline method to evaluate the delay time. Recall that the transaction flow is broadcast among IoT devices in the blockchain network. After finding the target IoT gateway device, the requester would use a routing protocol to send the relevant data to that device. Hence, the approach of separation of the transaction flow and the data flow theoretically can save time for the requester to send data. To demonstrate our theoretical analysis, we conduct the relevant experiments. The result is presented in Fig. 16. The delay time is directly proportional to the number of IoT gateway nodes. In other words, the delay time in the baseline method grows linearly as the number of IoT gateway increase. The linear growth explains that data broadcast can cause a huge amount of delay in a blockchain-based IoT environment. By considering scalability in an IoT environment, the delay time is unacceptable with thousands of IoT gateway devices. On the contrary, the consumed time is less in our framework. Since the data is forwarded by using a outing protocol, the number of IoT gateway nodes does not have a huge impact on the delay time. The main reason is that there is not much data to be carried in a transaction flow. On the other hand, the results of our experiment are not ideal, because the IoT gateway nodes only simulated up to 8 devices. In the real world, there are thousands of IoT gateway devices. The time difference between the baseline and our framework can be more oblivious. In order to obtain standard data, the experiment is executed 10 times to achieve an average value.

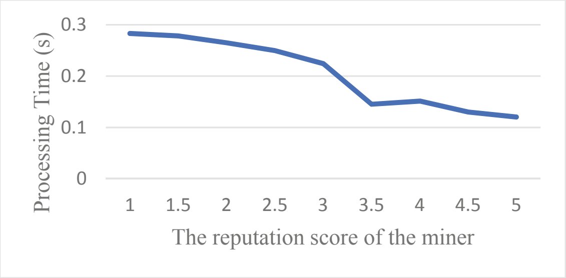
## 5.2 *Verification Time*

In the classical blockchain, a new block is verified among a few nodes. In our framework, the verification scheme is based on the reputation score and the public-key cryptosystem. Recall that in Sect. 3, we propose a reputation-based verification method. In this approach, the verifier firstly verifies the public key of each transaction. Then, verifiers check the signatures of each transaction based on the previous public key. Finally, the number of transactions that must be verified is gradually decreased with a higher reputation score for the miner. We also assume that at least half of the IoT gateways nodes are honest. We use the default settings and configurations of reputation scores in the experiment. The time that we evaluated is from the block generated to the block appended into the blockchain.

The baseline used in this experiment is that all transactions must be verified in each block. This approach provides an extremely secure environment since each transaction is validated. However, this method might be considered as an overcaution. By considering IoT devices, we need to balance the security and the performance of our framework. Therefore, we trade off security to gain better services in an IoT environment. However, we assume that the higher the reputation score of IoT gateway devices, the more honest they can be. It means that it is unlikely for an IoT gateway device, which has a reputation score of 5, to perform dishonest and malicious activities in our framework. Even though they may conduct some malicious activities, we still have the JC for protection. The JC can implement penalties for the abnormal and malicious IoT gateway nodes.

In this experiment, we compare the baseline method with the approach used in our framework. The result is shown in Fig. 17. In the baseline approach, there are 8 IoT gateway devices used in the experiment. Moreover, we still use 8 IoT devices to simulate our framework. The difference is that we did 9 experiments. In each experiment, we change the value of the reputation score of the miner from 1 to 5. In this way, the number of transactions to be verified is different in each experiment. From the results, the processing time in the baseline method is always 0.28310 s, because all transactions are verified by verifiers. This baseline cannot be shown in Fig. 17, because we have two different axes in this experiment. On the other hand, we use a reputation-based verification scheme to evaluate the processing time for a new block in our framework. The reputation score has a significant impact on the processing time in verification. With the lowest reputation score, each node needs to verify all the transactions. It takes the same time as in the baseline method. However, with the highest reputation score, it only causes 0.12036 s. The processing time in the highest reputation score is less than half of the processing time in the baseline

**Fig. 17** The verification processing time with different reputation score



method. Consequently, the processing time can be much less than the processing time in the baseline approach.

We only can say that the processing time in our framework is less than the processing time in the baseline method. However, we cannot analyse how much less in our framework because the miner is randomly selected in each round. It means that we cannot ensure that the IoT gateway node with the highest reputation score is selected to be the miner. In our framework, we assume that IoT gateway nodes, which pass the threshold score, are honest and trustworthy. For example, two IoT gateway nodes, which have reputation scores of 4.5 and 5 respectively, are both regarded as honest and fair. Therefore, we cannot provide an exact value for how much time our framework can save.

### *5.3 Transactions Per Second*

In the blockchain, transactions per second (TPS) means the number of transactions that can be processed per second in a network. The higher value of the TPS means that more events can be processed in a network. In this chapter, we need a relativity high TPS in our framework, because some IoT devices have strict time requirements. The classical blockchain has a fixed throughput. For example, Bitcoin has 7 TPS, and Ethereum has 25 TPS. The baseline in this simulation is the Ripple. We evaluate the TPS performance with different concurrent transactions. Later, we take the average value from different situations with different concurrent transactions. In case of any errors and defects in our system, the experiment is executed 10 times to achieve an average value.

As one of the high TPS consensus algorithms, Ripple is based on an FBFT method. The main purpose of this method is to reduce latency in the network. The TPS in Ripple is around 1500, which is a good candidate consensus algorithm in an IoT environment. However, the faulty nodes in Ripple can only be up to 20%. This means that most of the nodes must be honest and perform normally.

The TPS with different IoT gateway nodes is listed in Fig. 18. To achieve an accurate value in TPS, we implement our consensus algorithms into a different number of concurrent transactions. With different concurrent transactions, the final value of the TPS can be closed with the value in real-world scenarios. From our experiment, the average TPS can be reached around 1988. The overall performance of our framework is much better than the performance of other trust-based consensus algorithms, such as Ripple. The lowest TPS that we have achieved is 1978, while the TPS in Ripple is around 1500 transactions per second. Observe that the maximum difference between the highest value and the lowest value in TPS is only 62, which is acceptable when compared with the overall TPS in our framework. From another perspective, the trend of the TPS decreases as the number of IoT gateway increases. It explains that transactions need to be broadcasted among a blockchain network. In this case, it may cause more time for transactions to be appended into the blockchain. The experiment is executed 10 times to achieve an average value.

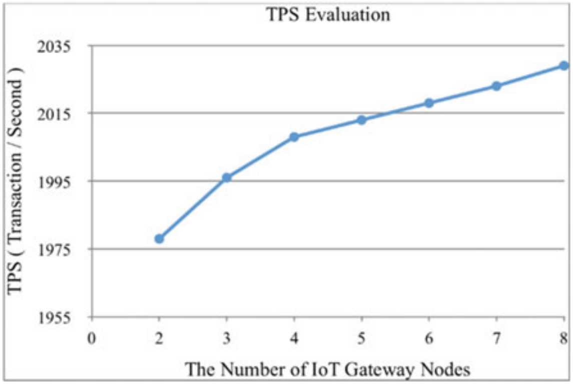


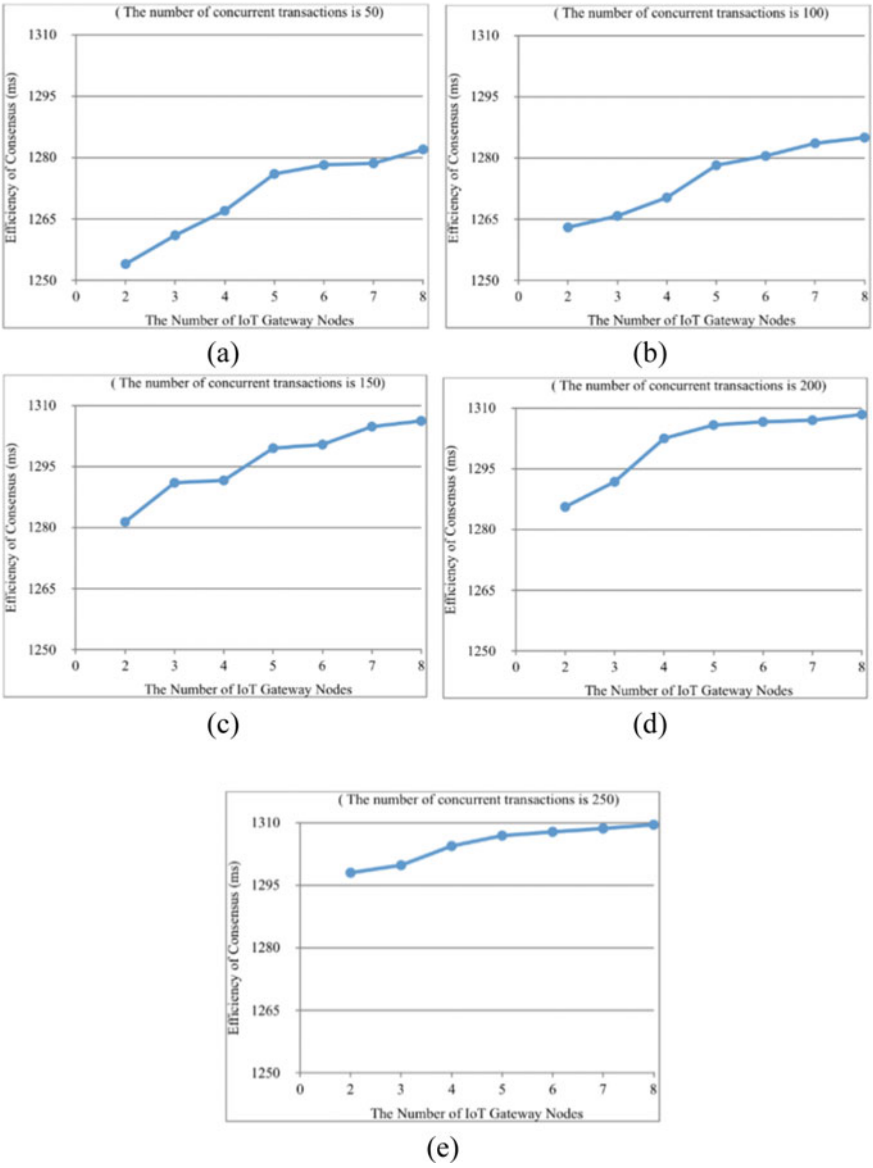
**Fig. 18** The performance of TPS

## 5.4 *Sybil Attack Evaluation*

In the previous section, we have discussed general information about the Sybil attack. It can cause tremendous damage to blockchain systems. For example, it can cause double-spending in Bitcoin-based systems. However, our framework can defend against this kind of attack based on our consensus algorithm. Firstly, we assume the worst case is that the proportion of honest IoT gateway nodes equals n/2 + 1, which n is the total number of the IoT gateway nodes. It means that there are 3 malicious nodes when there are 8 IoT gateway devices in our framework. Therefore, the minimum number of devices can be 3 in our experiment.

In this experiment, the malicious IoT gateway can create false transactions and blocks. Later, other malicious nodes agree on those transactions, which are generated by their fellows. We simulate that one of the malicious IoT gateway nodes can be the miner. It can create false transactions which include the permission of requests of access control policies. Other malicious IoT gateway nodes would agree on that block. Based on the settings and default configurations, we evaluate our framework with one metric, which is the efficiency to generate a new block. From the efficiency analysis, we can demonstrate that our framework can defend against Sybil attacks. Besides that, there is a slight impact on the efficiency to generate a new block. We implement our framework in different situations. The number of concurrent transactions increases from 50 to 250 with a step of 50.

In Fig. 19, it provides the efficiency to append a new block by using our framework. During the process, it contains the miner selection and the transaction verification. The efficiency only focuses on how much time is consumed in the miner selection and verification. The most time consumed in the miner selection is to identify the eligible potential miners. It doesn't need too much time once the threshold score is adjusted appropriately. In the verification, false transactions can be easily identified by checking corresponding public keys and signatures. Moreover, since we only verify a fraction of transactions in a new block, it saves a huge amount of time. In



(a) (b) (c) (d) (e)

**Fig. 19** The efficiency of our framework under Sybil attacks

order to demonstrate the stability of our framework, we use a different number of concurrent transactions to show consistency.

From each experiment, the efficiency maintains around the same time in each different situation. It means that our framework performs stable with different numbers of nodes and concurrent transactions. From the research [39], Zou et al.

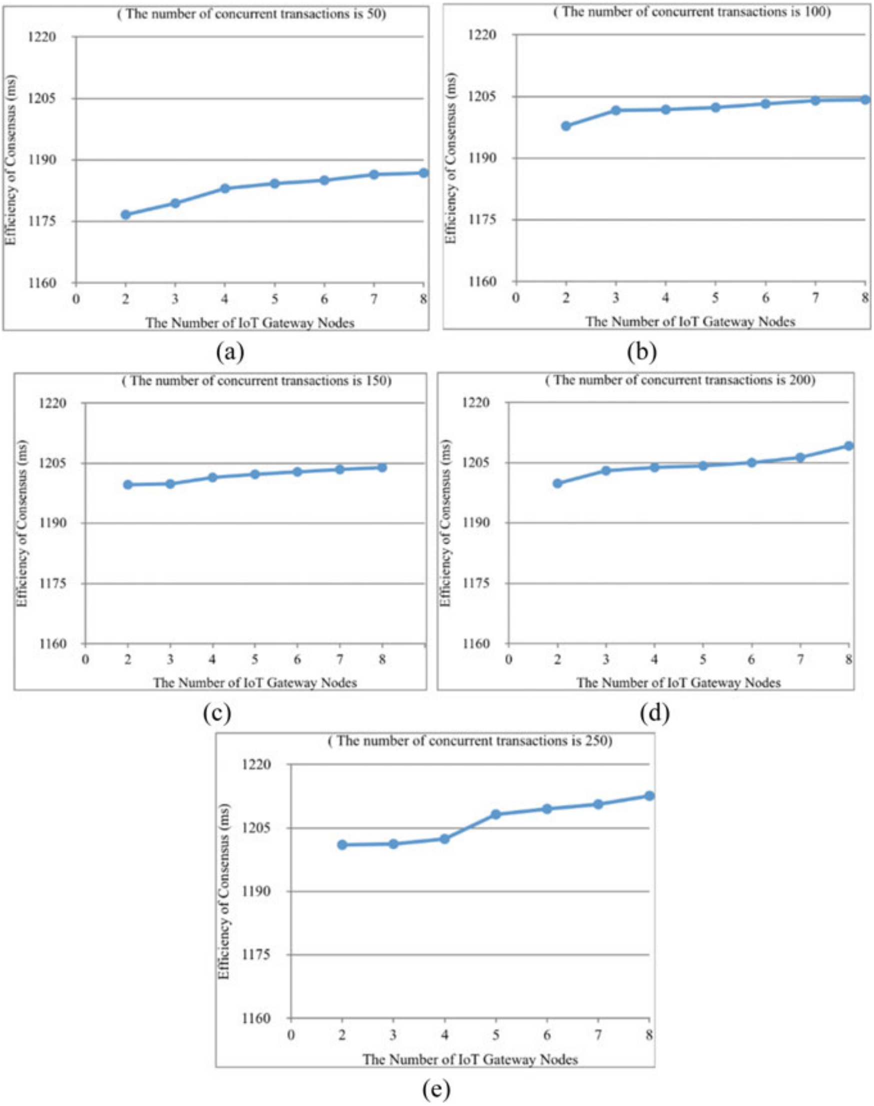
also tested their framework under Sybil attacks. Their experiments contain sub-network-based consensus protocol, joint consensus protocol and Proof of Trust (PoT) protocol. All the protocols, including their framework (PoT), have relatively low efficiency. They all achieve more than 2000 ms to generate a new block under Sybil attacks. Particularly, the Joint consensus protocol can take up to 6000 ms to generate a block. However, our framework can achieve 1274 ms on average. It demonstrates the efficacy of generating transactions and blocks under Sybil attacks in our framework.

In the real world, the value of concurrent transactions changes every second. There are many factors that can impact the value of concurrent transactions in a blockchain system. The experiment has been done 5 times to avoid any errors and defects from the computer. From Fig. 19, it provides the relationship between the number of IoT gateway nodes and the efficiency of consensus in a different number of concurrent transactions. In addition, it demonstrates that as the number of IoT gateway nodes increases, our framework needs more time to achieve an agreement. In each subgraph, there is a linear growth when the number of IoT gateway nodes increases. The main reason for this satiation is that it takes more time to select the miner in each round. It involves more verifiers when the number of IoT gateway nodes increases. Consequently, the consensus algorithm consumes more time when the number of IoT gateway nodes increases in our framework. Moreover, the number of concurrent transactions can influence the efficiency of the consensus. The average value of the efficiency of the consensus algorithm is different in each situation. From our experiments, they are 1270.85, 1275.27, 1296.41, 1301.05 and 1305 ms. It clearly shows that the efficiency of the consensus is directly proportional to the number of concurrent transactions. Since the blockchain network is based on a P2P network, transactions are broadcast in a network. The more transactions in a blockchain network, the more time is needed to be consumed for a transaction transmitted in the network. Besides that, after the miner creates a block, the block needs more time to broadcast to the verifiers in the network with a large number of concurrent transactions. Therefore, the efficiency of the consensus is related to the number of IoT gateway nodes and the number of concurrent transactions in a blockchain system.

### *5.5 DDoS Attack Evaluation*

The settings are quite similar to the configurations in Sect. 5.4. The only difference is the malicious activities that have been conducted in the experiment. The malicious nodes are also less than half of the participating nodes in the blockchain network. The malicious activity is that all dishonest IoT gateway nodes send requests to honest nodes. In this case, many resources have been occupied. It damages the honest services in an IoT environment. We let the malicious IoT gateway node sends 10 transactions per second to affect normal operation in our framework. Besides that, we evaluate this experiment by the efficiency to generate a new block.

In Fig. 20, it shows the time consumed to successfully generate a block under a DDoS attack in our framework. When facing a DDoS attack, our framework has


( The number of concurrent transactions is 50)

(a)

( The number of concurrent transactions is 100)

(b)

( The number of concurrent transactions is 150)

(c)

( The number of concurrent transactions is 200)

(d)

( The number of concurrent transactions is 250)

(e)

**Fig. 20** The efficiency of our framework under DDoS attacks

unique mechanisms to ensure security for our data. Recall that in Sect. 3.2, smart contracts can defend against such attacks in our framework. The malicious nodes can be recorded, and corresponding punishments issued. The entire process can be done fairly and rapidly. Compared with a similar framework, Zou et al. conducted similar experiments under collision attacks [34]. The consortium protocol has more efficiently to generate blocks, since it can ensure the behaviours of nodes are trustworthy. The main reason for such a circumstance is that each node has already completed an authentication check before it joins a consortium blockchain. Since our framework incorporates smart contracts, it needs to consume minimal time to append a block into the blockchain. In particular, the average time to generate a block is 1196 ms. In [39], the Proof-of-Trust (PoT) framework can generate a block between 3142.8 and 4785.4 ms. Compared to those two data, our framework can achieve half of the time consumed in PoT.

From Fig. 20, the number of concurrent transactions influences the efficiency of the consensus. There is a linear growth as the number of concurrent transactions increases. The main reason for this situation is that transactions need more time to be broadcast in our blockchain network. The more concurrent transactions in a blockchain network, the high possibility of causing network congestion. In this way, the average time for a consensus period should be longer for the blockchain system, which has a large number of concurrent transactions. In each of the graphs, we can see that the consensus algorithm takes more time in the network with more IoT gateway nodes. The main reason is that more nodes are involved in the consensus period. The generated block needs to be broadcast further in the blockchain network. Besides that, it needs more than half of the nodes to verify the transactions and blocks. It may take more time as the number of IoT gateway nodes increases.

## 6 Conclusion

The traditional approach of using the access control scheme has many limitations and challenges with IoT devices. Even though blockchain technology has provided benefits in IoT. there are still several significant limitations including dishonest participating nodes, complex consensus algorithms and latency overheads. To solve these challenges, we proposed lightweight blockchain-based trust management for access control in IoT. Our framework provides a novel trust management system that reduces unfairness and untrustworthiness in public blockchain platforms.

In this chapter, our framework has a smart contract-based access control system to manage participating parties in an IoT environment node. The main purpose of smart contracts is to create a fair and trustworthy environment. Moreover, the consensus algorithm does not solve any puzzle to append a block into the blockchain. The miner selection is based on the reputation score and sleep-time for each IoT device. The verification approach incorporates the reputation score of the miner, which significantly reduces processing time. Besides that, transactions that need to be validated are gradually decreased as the reputation score of the miner increases. We also present a case study, which illustrates the workflow of implementing smart contract-based access control. The functional analysis demonstrates that our framework can theoretically achieve validity, liveness, scalability, fairness, and security. Simulation results show that the proposed framework decreases delay time and processing time compared with classical blockchains. TPS in our framework is higher than other consensus algorithms that are used in IoT environments. In addition, the behaviours of each node are evaluated by our framework. The efficiency of the consensus algorithm has a slight influence under Sybil and DDoS attacks.

Overall, our framework brings a high level of security and anonymity to IoT users. We plan to improve on other aspects of our framework, which include the throughput, scalability, and trust execution environment. Besides that, we would like to implement our framework with a real-world platform. Thus, we can see the limitations and challenges in our framework. Lastly, we can explore our framework in other industries, which may include smart grids and vehicular networks.

## References

1. S. Pal, M. Hitchens, V. Varadharajan, Towards a secure access control architecture for the internet of things, in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)* (IEEE, 2017), pp. 219–222
2. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. **17**(4), 2347–2376 (2015)
3. R. Kandaswamy, D. Furlonger, Blockchain-based transformation (2018). https://www.gartner.com/en/doc/3869696-blockchain-based-transformation-a-gartner-trend-insight-report
4. H. Heinonen. Internet of things booming 15 trillion market (2020). https://towardsdatascience.com/internet-of-things-booming-15-trillion-market-88fde1da2113
5. I. Ullah, H. Zahid, F. Algarni, M.A. Khan, An access control scheme using heterogeneous signcryption for IoT environments. CMC-Comput. Mater. Contin. **70**(3), 4307–4321 (2022)
6. V. Suhendra, A survey on access control deployment, in *International Conference on Security Technology* (Springer, 2011), pp. 11–20
7. M. Ma, G. Shi, F. Li, Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario. IEEE Access **7**, 34045–34059 (2019)
8. A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: the blockchain model of cryptography and privacy-preserving smart contracts, in *2016 IEEE Symposium on Security and Privacy (SP)* (IEEE, 2016), pp. 839–858
9. K. Nguyen, S. Pal, Z. Jadidi, A. Dorri, R. Jurdak, A blockchain-enabled incentivised framework for cyber threat intelligence sharing in ICS (2021). arXiv:2112.00262
10. Q. Wang, J. Huang, S. Wang, Y. Chen, P. Zhang, L. He, A comparative study of blockchain consensus algorithms. J. Phys.: Conf. Ser. **1437**(1), 012007 (IOP Publishing, 2020)
11. E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes. IEEE Commun. Surv. Tutor. **7**(2), 72–93 (2005)
12. Z.-K. Zhang, M.C.Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, IoT security: ongoing challenges and research opportunities, in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications* (IEEE, 2014), pp. 230–234
13. S. Pal, Z. Jadidi, Protocol-based and hybrid access control for the IoT: approaches and research opportunities. Sensors **21**(20), 6832 (2021)
14. S. Pal, Z. Jadidi, Analysis of security issues and countermeasures for the industrial internet of things. Appl. Sci. **11**(20), 9393 (2021)
15. E. Bertino, N. Islam, Botnets and internet of things security. Computer **50**(2), 76–79 (2017)
16. C.J. D'Orazio, K.-K.R. Choo, L.T. Yang, Data exfiltration from internet of things devices: IOS devices as case studies. IEEE Internet Things J. **4**(2), 524–535 (2016)
17. S. Sahraoui, A. Bilami, Compressed and distributed host identity protocol for end-to-end security in the IoT, in *2014 International Conference on Next Generation Networks and Services (NGNS)* (IEEE, 2014), pp. 295–301

18. J. Liu, Y. Xiao, C.P. Chen, Authentication and access control in the internet of things, in *2012 32nd International Conference on Distributed Computing Systems Workshops* (IEEE, 2012), pp. 588–592
19. S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, A. Hill, On the design of a flexible delegation model for the Internet of Things using blockchain. IEEE Trans. Ind. Inf. **16**(5), 3521–3530 (2019)
20. S. Pal, M. Hitchens, V. Varadharajan, On the design of security mechanisms for the internet of things, in *2017 Eleventh International Conference on Sensing Technology (ICST)* (IEEE, 2017), pp. 1–6
21. S. Pal, M. Hitchens, T. Rabehaja, S. Mukhopadhyay, Security requirements for the internet of things: a systematic approach. Sensors **20**(20), 5897 (2020)
22. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A survey on access control in the age of internet of things. IEEE Internet Things J. **7**(6), 4682–4696 (2020)
23. D.D. Downs, J.R. Rub, K.C. Kung, C.S. Jordan, Issues in discretionary access control, in *1985 IEEE Symposium on Security and Privacy* (IEEE, 1985), pp. 208–208
24. E. Bertino, S. Jajodiat, P. Samarati,Enforcing mandatory access control in object bases, in *Security for Object-Oriented Systems* (Springer, 1994), pp. 96–116
25. D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST standard for role-based access control. ACM Trans. Inform. Syst. Secur. (TISSEC) **4**(3), 224–274 (2001)
26. P.A. Bonatti, P. Samarati, A uniform framework for regulating service access and information release on the web. J. Comput. Secur. **10**(3), 241–271 (2002)
27. A. Ouaddah, I. Bouij-Pasquier, A. Abou Elkalam, A.A. Ouahman, Security analysis and proposal of new access control model in the internet of thing, in *2015 International Conference on Electrical and Information Technologies (ICEIT)* (IEEE, 2015), pp. 30–35
28. I. Bouij-Pasquier, A. Abou El Kalam, A. A. Ouahman, M. De Montfort, A security framework for internet of things, in *International Conference on Cryptology and Network Security* (Springer, 2015), pp. 19–31
29. M.A. Khan, K. Salah, IoT security: review, blockchain solutions, and open challenges. Fut. Gener. Comput. Syst. **82**, 395–411 (2018)
30. S. Pal, M. Hitchens, V. Varadharajan,Access control for internet of things—Enabled assistive technologies: an architecture, challenges and requirements, in *Assistive Technology for the Elderly* (Elsevier, 2020), pp. 1–43
31. N. Chaudhry, M.M. Yousaf, Consensus algorithms in blockchain: comparative analysis, challenges and opportunities, in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)* (IEEE, 2018), pp. 54–63
32. C. Cachin, Architecture of the hyperledger blockchain fabric, in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310, no. 4 (Chicago, IL, 2016)
33. Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things. IEEE Internet Things J. **6**(2), 1594–1605 (2018)
34. A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: the case study of a smart home, in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (IEEE, 2017), pp. 618–623
35. D.D.F. Maesa, P. Mori, L. Ricci, Blockchain based access control, in *IFIP International Conference on Distributed Applications and Interoperable Systems* (Springer, 2017), pp. 206–220
36. A. Ramachandran, D. Kantarcioglu, Using blockchain and smart contracts for secure data provenance management (2017). arXiv:1709.10000
37. B. Shala, U. Trick, A. Lehmann, B. Ghita, S. Shiaeles, Blockchain-based trust communities for decentralized M2M application services, in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (Springer, 2018), pp. 62–73
38. M. Steinheimer, U. Trick, B. Ghita, W. Fuhrmann, Autonomous decentralised M2M application service provision, in *2017 Internet Technologies and Applications (ITA)* (IEEE, 2017), pp. 18–23
39. J. Zou, B. Ye, L. Qu, Y. Wang, M.A. Orgun, L. Li, A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. IEEE Trans. Serv. Comput. **12**(3), 429–445 (2018)

40. L. Bahri, S. Girdzijauskas, When trust saves energy: a reference framework for proof of trust (PoT) blockchains, in *Companion Proceedings of the The Web Conference 2018* (2018), pp. 1165–1169
41. B. Shala, U. Trick, A. Lehmann, B. Ghita, S. Shiaeles, Novel trust consensus protocol and blockchain-based trust evaluation system for M2M application services. Internet Things **7**, 100058 (2019)
42. H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, J. Kishigami, Blockchain contract: securing a blockchain applied to smart contracts, in *2016 IEEE International Conference on Consumer Electronics (ICCE)* (IEEE, 2016), pp. 467–468
43. F. Gai, B. Wang, W. Deng, W. Peng, Proof of reputation: a reputation-based consensus protocol for peer-to-peer network, in *International Conference on Database Systems for Advanced Applications* (Springer, 2018), pp. 666–681
44. E.K. Wang, Z. Liang, C.-M. Chen, S. Kumari, M.K. Khan, PoRX: a reputation incentive scheme for blockchain consensus of IIoT. Fut. Gener. Comput. Syst. **102**, 140–151 (2020)
45. J. Yu, D. Kozhaya, J. Decouchant, P. Esteves-Verissimo, Repucoin: your reputation is your power. IEEE Trans. Comput. **68**(8), 1225–1237 (2019)
46. N. Hajdarbegovic. Bitcoin miners ditch Ghash.io pool over fears of 51% attack (2021). https://www.coindesk.com/markets/2014/01/09/bitcoin-miners-ditch-ghashio-pool-over-fears-of-51-attack/
47. G.O. Karame, E. Androulaki, S. Capkun, Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (2012), pp. 906–917
48. J. Joshi, R. Mathew, A survey on attacks of bitcoin, in *International Conference on Computer Networks, Big Data and IoT* (Springer, 2018), pp. 953–959
49. J.R. Douceur, The sybil attack, in *International Workshop on Peer-to-Peer Systems* (Springer, 2002), pp. 251–260
50. Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, Z. Tian, Toward a comprehensive insight into the eclipse attacks of tor hidden services. IEEE Internet Things J. **6**(2), 1584–1593 (2018)
51. S. Wani, M. Imthiyas, H. Almohamedh, K.M. Alhamed, S. Almotairi, Y. Gulzar, Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight. Symmetry **13**(2), 227 (2021)
52. M.I. Mehar et al., Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack. J. Cases Inf. Technol. (JCIT) **21**(1), 19–32 (2019)

# Utilising K-Means Clustering and Naive Bayes for IoT Anomaly Detection: A Hybrid Approach

**Lincoln Best, Ernest Foo, and Hui Tian**

**Abstract** The proliferation of Internet of Things (IoT) devices means that they have increasingly become a viable target for malicious users. This has created a need for greater flexibility in anomaly detection algorithms that can work across multiple devices. The suggested algorithm will further ensure that our data remains secure from malicious users and potentially avoiding related real-world issues. This chapter suggests a potential alternative anomaly detection algorithm to be implemented within IoT systems that can be applied across different types of devices. This algorithm comprises both unsupervised and supervised areas of machine learning, utilising the strongest facets of each methodology. These are the speed of unsupervised, as well as the accuracy of supervised machine learning. The algorithm involves the initial unsupervised k-means clustering of attacks. The k-means clustering algorithm groups the data as either DDOS, backdoor, ransomware, worm, trojan, password, and normal and assigns them to their clusters. Next, the clusters are then used by the AdaBoosted Naïve Bayes supervised learning algorithm to teach itself which piece of data should be clustered to which specific type of attack. This increases the accuracy of the proposed algorithm by adding clustered data before the final classification step, ensuring a more accurate algorithm that can effectively classify attacks. The correct identification percentage scores for this proposed algorithm range from anywhere from 90 to 100%, as well as rating the proposed algorithm's accuracy, precision, and recall on different datasets. These high scores demonstrate an accurate, flexible, scalable and optimised algorithm that could potentially be utilised by different IoT devices, ensuring strong data integrity and privacy.

L. Best (✉) · E. Foo · H. Tian
School of Information and Communication Technology, Griffith University, Brisbane, QLD 4111, Australia
e-mail: lincoln.best@griffith.edu.au

E. Foo
e-mail: e.foo@griffith.edu.au

H. Tian
e-mail: hui.tian@griffith.edu.au

S. Pal et al. (eds.), *Secure and Trusted Cyber Physical Systems*, Smart Sensors, Measurement and Instrumentation 43, https://doi.org/10.1007/978-3-031-08270-2_7

**Keywords** Anomaly detection · Machine learning · IoT · AdaBoost · Naïve Bayes · K-means

# 1 Introduction

Detecting anomalies within Internet of Things (IoT) devices has become an increasingly important aspect of cybersecurity due to the growth in prevalence of these attacks [1]. This growth has been fuelled by the current covid-19 pandemic, as large numbers of employees are working remotely from home. The increase in time spent at home means IoT devices are therefore used more, generating larger amounts of data, and subsequently becoming more attractive to malicious users. With the subsequent attractiveness of IoT devices increasing, a greater need for information security and specifically anomaly detection is required (AD).

## *1.1 Background*

On a conceptual level, the IoT is an interconnected ecosystem that is populated by devices that have computing and networking capabilities embedded within the object [2]. IoT devices aim to add value as well as personalise the user's experiences and interactions with various "things". IoT devices enable large-scale technological advancements in many different areas such as agriculture, smart cities, health and fitness, traffic, retail, and logistics. The IoT is often seen as a global infrastructure that enables connectivity between the cyber and physical worlds based on existing structures, frameworks as well as previous, more basic IoT devices [2]. As IoT devices rely on a connection to each other and the internet, this means that they are potential targets for malicious users, and therefore require measures to prevent and detect intrusions within a connected network.

AD within networking is utilised to ascertain behaviours deviating from the norm [3]. An example of networking anomaly detection's actions would be the identification of suspicious networking packets from IP addresses flagged as malicious, or an abnormal amount of traffic sent to and from internal or external networks. This anomalous behaviour often takes place within or against traditional components of a network. These traditional networking components (firewall, router and switch) rely on AD algorithms, however, networking for IoT is more difficult, as their security requirements differ due to the size, scalability and resources of connected devices [4].

The current IoT security measures are based on the principles of confidentiality, integrity, authorisation and availability. These measures include specific protocols for connecting and communicating with and through the internet, as well as machine learning algorithms for anomaly detection [5].

Examples of current security protocols for IoT are MQTT, CoAP, and AMQP. The MQTT (Message Queuing Telemetry Transport Protocol) model operates similarly to the HTTP, request-response operation as opposed to the MQTT mode that uses a publish-subscribe model. The publish-subscribe model operates by the publisher, first collecting the entity's data from different sensors. The subscriber of this specifically subscribes to the sensors data which is displayed to the end-user. The third stakeholder within MQTT is the broker, which operates as the go-between, delivering the data from the publisher to the subscriber [6]. CoAP (Constrained Application Protocol) operates like the MQTT model however it relies on a URI (universal resource identifier). The CoAP model publishes data gathered to the URI, and the subscriber (user) subscribes to the resource indicated by the URI. When new data is published to the URI, the other users are notified. AMPQ (Advanced message queuing protocol) operates by utilising both the publish-subscribe and the request-response models. The AMPQ model communicates by requesting the user or publisher create and then broadcast that exchange. Subsequent communications take place by either the broadcaster or user utilising that name within the exchanges. At the same time, this is occurring, the user must create a queue and attach it to the exchange, with messages then matched to the queue, in order for authentication to occur. These security protocols operate concurrently with different AD methods to secure information and prevent malicious users from gaining access to IoT devices [].

There are numerous current methods employed by IoT devices to ensure data and device integrity. These include but are not limited to authentication and access control, attack detection and mitigation, anomaly intrusion and detection and malware analysis [7]. These methods rely on different machine learning (ML) techniques to operate. Authentication and access control rely on artificial neural networks (ANN) and long short-term memory (LSTM). Attack detection and mitigation require a support vector machine (SVM), as well as deep learning autoencoders and K-nearest neighbours (KNN). Malware analysis can use recurrent neural networks (RNN), principal component analysis (PCA) and convolutional neural networks (CNN).

Intrusion and anomaly detection algorithms are sometimes composed of k-means (KM) clustering, decision trees (DT), and Naive Bayes (NB) [7]. Unsupervised machine learning (UML) algorithms have greater flexibility regarding data absorption and require far less manual handling than the supervised machine learning (SML) method [8].

Combining both UML and SML algorithms gives birth to hybrid machine learning (HML). HML involves utilising information associated with both ML strands and their strengths, whilst simultaneously minimising their weaknesses. An example of this is if there is an issue with a classification problem associated with SML, then using additional data points from UML could help with the labelling. Conversely, utilising a SML algorithm to assist a clustering algorithm with increased data point knowledge would also be a way to minimise weaknesses associated with an UML algorithm [9].

AD methods operate concurrently on IoT smart devices, utilising abovementioned ML algorithms. IoT devices generate different types of data, these include but are not limited to network and sensor data. This chapter will utilise telemetry/measurement data from sensors placed within several different IoT devices. This is done to further examine the usage of sensor data to detect anomalies within IoT devices.

## *1.2 Motivation of the Research*

This chapter seeks to answer the overarching question:

(1) How much better would a hybrid machine learning algorithm comprised, of k-means clustering and Naive Bayes be than traditional SML algorithms when it comes to AD within IoT sensor data?

    To answer the overarching question, there are several sub-questions featured within this chapter. Answering these sub-questions will allow for a complete analysis of the proposed algorithm. The sub-questions relate to accuracy, speed, scalability and flexibility are listed below:

(2) Does the proposed algorithm have higher accuracy, precision and recall scores than traditional SML methods?

(3) Does the proposed algorithm have faster train and test times than the traditional SML algorithms?

(4) Does the proposed algorithm maintain its strength as it operates on larger datasets?

(5) Is the proposed algorithm able to be applied to different types of IoT devices?

    Answering the questions relating to speed and accuracy of the proposed algorithm also subsequently raises another question that must be answered within this chapter. This question discusses the trade-off between speed and accuracy. This can be best phrased as:

(6) Does removing the highest-ranked subset of attributes to the type of anomaly, negatively impact the strength or speed of the algorithm?

The crux of the question is if you have an AD algorithm that is 90% accurate but takes 3 min to operate, versus an algorithm that is 87% accurate, that takes 30 s to predict, which algorithm is the better choice for the device? And would it be possible to find a way to increase efficiency by deleting data points whilst maintaining accuracy.

Answering the above questions will allow this chapter to assess whether the HML k-means and Naïve Bayes algorithm is indeed a viable alternative to traditional SML. This answer is based on the notion that having an AD algorithm that is slow, inaccurate, inflexible and unscalable would not be an effective use of resources within an IoT device.

### 1.3 Chapter Overview

The below chapters will be further broken down respectively: related work, proposed algorithm, evaluation, results, discussion, and conclusion and future work. The related work will focus on discussing previous literature including providing a brief overview of IoT sensors, contrasting IoT networking data, anomalies, anomaly detection, both strands of machine learning, as well as then going on to discuss the usage of KM and NB HML algorithms within academia. This chapter will then discuss how these algorithms detect anomalies and the subsequent different types of anomalies. After the related work, the new algorithm section details the requirements of the proposed algorithm, and how this proposed algorithm answers the research questions. The evaluation of the algorithm involves the discussion of the type of method used, as well as the subsequent hypothesis, the metrics for measurement as well as the process involved. The datasets will also be detailed including their components, sizes, types of IoT devices that were tested as well as different anomalies within each dataset. The results section will address each research objective and question, as well as discuss the advantages of new algorithms and their trade-offs. The discussion section will review the research questions and give specific insights. Lastly, the conclusion and future work section will summarise the entirety of this chapter, as well as provide any potential further work to be explored.

## 2 Related Work

This chapter will present what other authors have discussed regarding IoT, before going on to discuss sensor data, IoT anomalies and the subsequent different types, anomaly detection and machine learning (both supervised and unsupervised algorithms). Lastly, this chapter will then discuss hybrid ML before then giving an in-depth review of the many uses of HML algorithms, and then finishing with discussing the uses of k-means and or Naive Bayes within AD algorithms, as well as highlighting the gaps within research.

### 2.1 IoT Overview

Kassab and Darabkh [10] give an overview of IoT by first suggesting that there are issues within IoT devices. These issues are interoperability, scalability, resource scarcity and security. Kassab and Darabkh [10] begin by discussing the issue of interoperability, which is that different vendors devices might not work together, hence why the characteristic of interoperability is necessary. This can be achieved by ensuring that each protocol and sensor permits other different vendors from reading and accessing their accumulated data. As there are billions of devices within the IoT environment, the amount of data that is generated is large, the applications within need to be designed with the ability to be scalable enough to process the generated data. Kassab and Darabkh [10] discuss resource scarcity. Resource scarcity is the issue that relates to smart devices and the idea that they are resource-constrained, meaning that they are limited by computation and energy requirements. Finally, they discuss security. Security is arguably the most integral architectural conceptual issue within the IoT sphere. Security in this case refers to the intrusion prevention and anomaly detection within the IoT and their associated devices.

## *2.2 Contrasting IoT and Standard Networking Data*

There are several differences between IoT sensor and standard networking data. Alsaedi et al. [11] suggests that the current networking anomaly detection datasets, which are based on current systems primarily contain packet-level and flow-level information. This data is handy in detecting attacks on the network but fails to address the issue of attacks that specifically aim to change sensor data or manipulate IoT devices. Standard networking data found in basic datasets such as NSL-KDD does not contain information gathered from the actual sensors. Data gathered from the sensors, is relatively simple in contrast to the packet-level and flow level information previously mentioned [11]. Alsaedi et al. [11] suggest that this is a major gap in academia right now and that they attempting to address this.

## *2.3 Anomalies*

Quek et al. [12] discuss anomalies by first giving a base definition. According to the authors, anomalies are simply a deviation of normal conditions from the operating paradigm. They then go on to state that the two main assumptions regarding anomalies are that they occur rarely and that they are distinguishable from normal data. Looking specifically at the concept of anomalies within the IoT space, this typically refers to malicious incursions into the network or the IoT device [13]. Sahu and Mukherjee [13] then continue, stating that as there many different types of IoT devices within the ecosystem, therefore logically the types of anomalies also vary. This then brings the authors to discussing zero-day attacks, stating that zero-day attacks (ZDA) are numerous, and in effect are unknown anomalies that can be found within IoT devices. A ZDA refers to an event of involving different types of anomalies, including but not limited to denial of service, malicious control, malicious operation, scan and spying attacks. Sahu et al. then go on to list the types of anomalies, as well as background information and examples. They state that denial of service (DoS) attacks are found within traditional networking incursions. DoS attacks operate by making the required service crash or become unresponsive. This is done by ensuring that there are no allocated resources left to respond to the constant requests. The data type probing anomaly occurs when the device receives data that has been changed, an example of this is if a sensor is expecting an integer, but instead receives a string. Malicious control refers to another unintended user attempting to gain control of the network traffic. Malicious operation occurs when the original activity is obfuscated and hidden. Scanning anomalies refer to when a malicious user intends to replicate the client or server credentials to gain access to user data. The spying anomaly involves refers to eavesdropping on specific areas of the network or devices, to discover sensitive information [13].

## *2.4 Anomalies Within This Chapter*

The types of anomalies that are found within this chapter are scanning, distributed denial of service (DDoS), ransomware, backdoor, injection attack, cross-site scripting (XSS) and password cracking attacks. Alsaedi et al. [11] describes the types of attacks in detail. The authors start by stating that scanning attacks refer to the first step a malicious user takes to gain access to the network. Scanning attacks involve information gathering, targeting areas of vulnerability such as open ports and available services. DDoS attacks involve the malicious user flooding the victim with requests to disrupt access to services. They are often launched by many compromised devices known as botnets or bots. These compromised devices flood the target, often overwhelming their memory and bandwidth. This is particularly dangerous for IoT devices as they have limited computational power and storage capacity. Ransomware attacks are often malware-based and usually operate by denying the user access to a system or specific services. The malicious hackers will then sell decryption software back to the victims in exchange for returned access to their system. Backdoor is a passive type of attack, in which the bad actor will attempt to gain access to the victim's system remotely, usually with malware. These compromised systems often perform part of a botnet to launch DDoS attacks. Injection attacks involve executing snippets of malicious code or data into targeted applications. The injection attack can manipulate telemetry data and control commands, in order to disrupt normal operations. XSS attempts to operate malicious code on a web server that is connected to the targeted device. The XSS allows the malicious user to inject scripts of coding which can then compromise authentication procedures between different devices, and the webserver. The password cracking attack occurs when an attacker uses methods to guess a password and subsequently gain access to the adversarial system. This type of attack can allow attackers to bypass authentication methods [11].

## *2.5 Anomaly Detection*

Tsai et al. [14] state that anomaly detection within IoT comprises different areas, with differing methodologies. The authors then discuss the differing anomaly detection methodologies. The first anomaly detection method is classifying either signature-based or semantic-based. Signature-based AD refers to detecting attacks through threats or signatures that are already known. Anomaly-based detection schemes operate by employing statistical, machine or protocol-specific information and then building a model featuring legitimate traffic. This model is then used as a reference point to classify either normal or abnormal traffic. There are also hybrid systems that combine both detection and classification methods according to [15]. Other anomaly detection methods can occur in real or non-real time, real time referring to occurring synchronously, and non-real time meaning asynchronous. Anomalies can be found within either the actual network flows of IoT devices or by examining sensor information [14]. Some of the common types of machine learning and anomaly detection algorithms are discussed below.

## *2.6 Machine Learning*

Bengio et al. [16] discusses AD machine learning methods and then breaks them down into SML and UML. For SML, the training set functions as samples of input data points and are utilised in conjunction with corresponding, appropriate target vectors (labels). This contrasts with unsupervised learning, which does not require the use of labels.

The main objective of SML is to learn how to predict the output data from a given input vector. SML can then use classification with the target labels to examine a number of discrete categories within the data set and assign them names [16]. This chapter will detail SML algorithms, and then discuss UML in the next section.

## *2.7 Supervised Learning*

The SML algorithms that will be discussed below are K-Nearest Neighbours (KNN), Naïve Bayes (NB) and Random Forest (RF). These are chosen as they will be the traditional SML algorithms that our proposed algorithm will be tested against.

**K-Nearest Neighbours**. Mahdavinejad et al. [17] go into detail regarding KNN. They suggest that the main goal of KNN is to classify a new, discrete data point by finding the K-given data points in the training set. This is done by examining the data points closest to the input or feature space. To find the KNN, a measure of distance metric (Euclidean distance, $L\infty$ norm, angle, Mahalanobis or hamming distance) must be utilised. Jagdish et al. (2005) state that to solve the problem, the new data point (input vector) is seen as $x$, its K Nearest Neighbours by Nk($X$), the predicted class label for $x$ by $y$, and the specific class variable by $t$ (a discrete random variable). Furthermore, 1(.) denotes the indicator function: $1(s) = 1$ if s is true and $1(s) = 0$ if not. The input data point (x) will be generated by the mode of its neighbour's labels (Jagdish et al. 2005).

The authors then go on to depict KNN in the following way:

KNN is mathematically depicted as Eq. 1:

$$p(t = c|x, K) = {}^{1}\!/_{Ek} \sum i \in Nk(x)\ 1(ti = c)$$
$$y = \arg\max p(t = c|x, K)$$

Like every other type of ML algorithm, there are both pros and cons. A downside of KNN is that it requires the storing of the entire dataset, meaning the more data there is, the more that must be stored by the algorithm. This means that it is not a strong choice for larger datasets and is not as scalable as others.

**Naïve Bayes**. Zhang [18] discusses NB. They start by stating that NB's primary function is to apply the Bayes theorem with the naive assumption of independence between the features (attributes) of $z$ when given the class variable $t$.

Zhang [18] starts by denoting the input vector $z = (Z1,\ldots Z_M)$.

When applying the Bayes theorem, the below equation is used:

$$p(t = c)|Z_1 \ldots Z_M) = \frac{p(Z_1 \ldots Z_M|t = c)p_{(t=c)}}{p(t = c|Z_1 \ldots Z_M)}$$

Then subsequently integrate the naïve independence as well as the subsequent simplifications, which leaves us with:

$$p(t = c|Z\ _1, \ldots Z_M = \infty p(t = c)\Pi_{J=1}^{M} p\big(Z_j|t = c\big)$$

The way in which the classification takes place is shown below.

$$y = \arg m_c axp(t = c)\Pi_{J=1}^{M} p\big(Z_j|t = c\big)$$

It should be noted that $y$ shows the predicted class label for z. There are also different types of classifiers using different methods of distribution to estimate $p(t = c)$. NB already has a strong foundation within the AD sector, its uses already include spam filtering and text classifications [18]

**Random Forest**. Wang et al. [19] discuss the basic methodology of RF. They start by discussing the positives of RF and suggesting that it has a strong ability to handle various types of data, and already has a wide area of application within academia. Wang et al. [19] then describe the algorithm by first stating the assumptions. These assumptions are that a data set is annotated as $Dn$ with $n$ being the instances $(X, Y)$, it should also be noted that, $X \in \mathbb{R}D$. This approach combines numerous decision trees, independently trained to form a forest. It should also be noted that each tree is a partition of the data space. This means that as $\mathbb{R}D$ is the full set of data, then a leaf is a partitioned subsection of the entire dataset, and each node corresponds to a cell of data space. The methodology of the RF is presented below.

1. At the start of the tree construction, n sample points are taken from the *Dn* dataset. Only these samples are used to construct the tree.
2. Tree node *mtry* features (*mtry* < D) are then randomly sampled from the original D datasets. These samples are then used for the selection of the splitting point. After one split has occurred, the algorithm continues to split repeatedly until the stopping condition is met.
3. RF's then average the result from each tree [19].

## 2.8 Unsupervised Learning

Usama et al. [8] discuss UML generally before going onto data clustering. They start by stating that UML allows for the analysis of raw data, helping by generating insights into unlabelled data. UML has many different applications within the ML sphere. UML algorithms are utilised in areas of speech recognition and computer vision. Furthermore, UML's flexibility and scalability are seen as some of their key strengths. Due to these strengths, it could be suggested that UML could be applied to areas within network management, monitoring, and data optimisation. UML techniques can be divided into different sections, these include but are not limited to hierarchical learning, data clustering, latent variable models, dimensionality reduction techniques and outlier detection [8]. This chapter will specifically discuss data clustering below.

## 2.9 Data Clustering

Usama et al. [8] states that data clustering encompasses the organisation of data into natural, meaningful groups (clusters) based on the high similarity between different features. Clustering, therefore, attempts to find hidden patterns within the input, unlabelled vectors. The clusters are also organised in such a way that promotes high intra-cluster and low inter-cluster similarity [8]. The authors then suggest that clustering is widely applied to many different disciplines, these include but are not limited to ML, data mining, network analysis, pattern recognition, and AD. Data clustering can be further broken down into 3 areas, hierarchical clustering, Bayesian clustering, and partitional clustering. This chapter will look specifically at partitional clustering as it precedes K-means (KM) clustering, the UML algorithm used.

## 2.10 Partional Clustering

Usama et al. [8] discuss partitional clustering generally and then go on to discuss the pros of this method. They start by stating that clustering is a method of organising data into a set of disjointed clusters. Partitional clustering has an advantage over other types of anomaly detection algorithms in that they can incorporate knowledge relating to the size of clusters, by relying on the specified distance functions. These functions ensure accurate data shape generation within the various types of clustering algorithms. Frigui [20] mentions that there are several drawbacks to partitional clustering. These are the difficulty in determining the number of clusters, the predisposition to being negatively impacted by outliers and data noise, and the issue of cluster initialisation. Partitional clustering can be further broken down into K-medoids, expectation means, and k-means. As K-means was the chosen partitional clustering type, this is the algorithm that will be discussed below.

### 2.10.1 K Means Clustering

Zhao et al. [21] start by giving a general overview of the KM clustering algorithm. Zhao et al. [21] then state that KM clustering is defined as an iterative expectation maximisation approach. It operates by including 3 steps. The first step is to initialise the $k$ cluster centroids, the next is to assign each sample collected to its closest centroid, and the last step is to reorganise the cluster centroids with the assignments computed in step 2, and then repeating step 2 until convergence is met. These steps are repeated until the centroids of the clusters do not change between consecutive iterative rounds. Zhao et al. then go on to display the clustering procedure. This is done by stating that $\{x_{\mathrm{I}} \in R^d\}$ $i = 1\ldots n$ are samples that are required to be clustered and $C\ldots k \in R^d$ is the cluster centroids. The above function represents the iterative and clustering steps. Before the specific KM algorithm is depicted, there are several rules that need to be specified.

Qi et al. [22] describe the KM algorithm. This was done by first stating several assumptions. One assumption is that the given dataset is denoted as $D$, and then $D = \{pi|i = 1,\ldots, n\}$, pi found in d-dimensional space. The first step (seeding) begins by selecting $k$ clusters, by minimising the sum of squared errors (SSE). The first equation depicted shows the operation of the KM clustering algorithm. This algorithm is found below:

$$\mathrm{SSE} = \sum_{\mathrm{j=i}}^{\mathrm{k}} \sum_{\mathrm{i=1}}^{\mathrm{n}} \delta_{\mathrm{ij}} \left\| \mathrm{p_i} - \mathrm{m_j} \right\|^2$$

$$\left(\delta_{ij} = 1 \text{ if } p_i \in C_j \text{ and } 0 \text{ otherwise}\right)$$

It should be noted that where $\|p_i - m_j\|$ is shown, this indicates the distance between the point $p_i$ and the cluster $C_j$, as well as its cluster centre $m_j$. This is shown below in the subsequent equation.

$$m_j = \frac{\sum_{pi} \in C_j P_i}{\left|C_j\right|}$$

Qi et al. [22]. The combination of the previous supervised and unsupervised machine learning algorithms leads to the discussion regarding HML.

### 2.10.2 Hybrid Machine Learning

Li et al. [23] discuss HML algorithms. The authors suggest that these algorithms consist of two forms of machine learning, unsupervised and supervised. Both unsupervised and supervised machine learning algorithms have strengths and weaknesses. Li et al. [23] suggests that utilising a combination of these two types is an effective way of combating the other's weaknesses. A popular way is to utilise one UML algorithm as the data aggregator, and then another one as the classifier. This is what has been proposed, a clustering algorithm to gather data, and then a supervised classifier to classify the data.

After the basic overview of IoT and the discussion of types of ML approaches, this chapter will now detail the potential uses of HML in the papers below.

### 2.10.3 Uses for Hybrid Machine Learning Algorithms

One potential use for HML refers to the integration of KM and NB for smart air conditioning (AC) monitoring and control in WSAN networks, as proposed by Kristianto et al. [24]. Kristianto et al. [24] describes the combination of KM and NB, and the way it operates. This is done by utilising the classification of the NB algorithm to determine the operation of the air conditioner units (AC). The sensors generate the unsupervised dataset, which is then clustered and formed into a supervised dataset. This supervised dataset is then passed on to the NB classifier and the subsequent instructions are then passed onto the controlling server. New data is then assigned to the specific clusters based on the previously NB classifications of previously collected data. After that data is received by the server, it is then passed onto the remote actuator, which then determines the operation of the AC. Kristianto et al. [24] then show that their combination, in terms of results, is rated by accuracy, precision, recall and error rate with 90%, 83%, 100% and 10% respectively. However, the conclusion presented is basic and suggests that merely because they have strong scores, their algorithm is effective.

Wayahdi et al. [25] suggests a combination of KM and a NB classifier to classify an image. This is done by attributing numbers to sections of an image based on the characteristics and statistics of the picture, as well as the hue, saturation, red, green, blue, kurtosis and skewness. Wayahdi et al. [25] provided an example and stated that this was first done with an image of a banana which was then resized to 100 × 100 pixels. Then the image is extracted into different numeric characteristics, based on the attributes. The grouping occurs using KM clustering. The classification via NB is repeated for different images each with different centroids. The results for this state that a total correct percentage of 85% was found. It could be suggested that the 85% accuracy rate could be fractionally low, particularly when using an SML algorithm. [25].

Ali et al. [26] suggests a way in which to utilise KM and NB as well as feature selection in text document categorisation. The hybrid machine learning algorithm would be utilised by first pre-processing the documents before clustering, by removing redundant and duplicate words, question marks and conjunctions. Next, the feature selection phase refers to the operation of the proposed model that involves further pruning. After this, the KM algorithm is called, which then calculates centroids of clusters, the clusters themselves as well as the minimum distance function. The minimum distance function denotes which features are allocated to the nearest K cluster. Next, the optimisation by the NB algorithm creates a specific cluster according to the predicted probabilities. The optimisation process continues as new centroids are created with new documents; this keeps occurring till all the allocated documents are clustered. The results of this algorithm suggest that the combined KM based NB is an accurate algorithm over the 4 chosen datasets. This is evident as for the first dataset, the proposed algorithm received 91.60% purity and 72.20% entropy for the proposed model compared to the 86.80% purity score and the 77.00% entropy score. Entropy refers to the measure of quality for clustering. However, the authors chose how many k-clusters by trial and error, they did not demonstrate any method. This could have been improved by mathematically suggesting the strongest possibility or using an expectation means algorithm and automatically assigning the number of clusters as per the features within the dataset.

Fadhil [27] proposed an algorithm that operates on the hybrid KM and NB systems to predict the performance of an employee. In the first phase, the KM algorithm begins the clustering process to determine the training data. This data includes classes (excellent, very good, good, average, and bad). The Euclidean distance, using the class data, measures the distance between data and the first centroid of the algorithm. After the initial centroid initialisation and Euclidean distance measurement, calculations occur, the class for every cluster is then analysed, and the average data for each variable within said cluster is found. This keeps occurring till all the data has been clustered. The average data score for each variable is compared to the centroid and if it is not equal to the centroid's value, then the distance calculation is repeated until the average data is equal to the value of the centroid for each variable. After the value has been chosen, the NB classifier is used. With the best results of each centroid value being output to the user. A critique that can be seen is that this algorithm relies heavily on there being no outliers within the data, as the centroid value of this algorithm repeats itself till the centroid value is the same. As KM is outlier sensitive, it could be seen that any outlier or null data would affect the centroid, making the algorithm keep repeating the initialisation stage [27]. This appears to be a systemic floor within the methodology and could be addressed by firstly pre-processing and removing the obvious outliers or null values. This chapter will now discuss using KM and NB for AD within the field of networking. These efforts will be discussed, reviewed, and critiqued below.

### 2.10.4 Hybrid AD Algorithms Using KM and Naïve Bayes

There are several HML AD systems that have been proposed. One such AD system is that of a hybrid KM clustering and NB classification technique. For this literature review, only the KM and NB will be assessed. This AD system was suggested by authors [28].

Bagui et al. [28] provide an overview of the way the experiment is designed and then discuss the results and any issues that have arisen. Firstly, the authors state that the algorithm operates by ingesting 8000 random records from the UNSW-NB15 dataset. Feature selection is then performed using the KM clustering and Correlation based Feature Selection (CFS). CFS evaluates the benefits of each subset. This was run on each attack family (fuzzers, analysis, backdoor, dos, exploits, generics, reconnaissance, shellcode and worms). Once the features were selected for each of the attack families, the classification step began. The classification step involves the usage of 2 different algorithms, the NB and the J48 decision tree.

The results showed that NB produced the best rates of classification, coming in with 80.03%, 90.66%, 90.02%, 92.97%, 46.70%, 92.61%, 71.42%, 75.24% and 99% for fuzzers, analysis, backdoor, dos, exploits, generics, reconnaissance, shellcode and worms respectively with CFS. Contrasted to without CFS, the scores were 57%, 74%, 66%, 66%, 56.69%, 83%, 65%, 72%, and 84 respectively. Bagui et al. [28] found that there were substantial increases in accuracy for the detection rate related to the use of CFS. The proposed algorithm differs from the authors algorithm in several ways. The main difference is that feature selection is not at the algorithm. Instead, we are using CFS to rank the feature with the highest correlation to the label, and then using that ranking to remove the highest ranked feature. This is done to see if the training and testing time can be improved whilst maintaining accuracy. This is due in large to the data we are analysing, and the subsequent implementation methods. Since our data is not as complex, adding in feature selection would be redundant. Our algorithm is designed to be used on simpler data gathered from IoT sensors. This allows for a more agile and less processing intensive approach, with comparable if not more accurate scores.

Bhatt and Thakker [29] suggested an algorithm to aid in the removal of botnet attacks using an ensemble classifier within IoT devices. Bhatt and Thakker [29] suggested collecting hacker activity patterns from the IoT devices as opposed to traditional usage of network statistics. Next, the modelling of the attacking information takes place within a tree-based structure by stacking the classifier. Lastly, the attacks are then clustered by a protein similarity algorithm (PROSIMA). The similarities to the proposed algorithm are minor, in so far as this utilises a clustering algorithm, AdAboosting, as well as a classifier and IoT device data. It should be noted that that algorithm does in fact involve the use of feature selection and is more complex in nature. As mentioned above, our proposed algorithm appears to be less complicated in nature, which fits with the design ethos of speediness, due to the resource-constrained nature of IoT devices [29].

Om and Kundu [30] proposed a hybrid intrusion detection system that combines KM and two additional classification algorithms, KNN and NB. It consists of feature selection based on entropy evaluation operating on the KD-99 dataset [30]. Om and Kundu [30] go on to discuss the method of operation for their proposed NB intrusion detection system (IDS). The IDS starts with first applying KM clustering to the dataset, specifying the number of clusters into either normal or anomalous clusters. The number of clusters is set to 5 (user 2 root, remote to local, probe DoS and normal). The data is then separated into two parts, one part for testing and the other for training. In the training phase, the labelled records are assigned to the K-Nearest Neighbour. The KNN is then trained on these. The subsequent rest of the data is then passed through the KNN classifier, it should also be noted that this method involves feature selection. This algorithm operates similarly to the proposed algorithm, with the clustering into classification. This delivered strong results however, there was no mention of the training and testing time. Moreover, as KNN needs to ingest the entire dataset, this further reinforces the theory that this algorithm was particularly slow in comparison to others.

Sharma et al. [41] suggest an improved intrusion detection technique based on KM clustering via the usage of NB as a classifier. Sharma et al. [41] state that the algorithm consists of several steps. Firstly, it begins by pre-processing (feature selection) and normalisation. Next, the KM clustering algorithm is run, followed by the classification via NB. After this occurs, the testing and validation of the performance take place, with the results displayed. Once again, this model relies on the use of feature selection before the algorithm occurs, to assist with getting higher accuracy rankings. Furthermore, they do not utilise any boosting methods. This could potentially increase their accuracy, therefore reducing bias and variance and allowing for a stronger overall algorithm [41]. It should also be noted that Sharma et al. suggest that the design of their algorithm is meant to be a general algorithm, however, the results suggest a different picture. The results demonstrate that it is accurate to a degree, however, it generates more false positives in some areas, as well as being less accurate for 2 out of 5 types of attack then the baseline AD algorithms. This means that the aim of this algorithm is largely unmet, as only 2 attack types were detected consistently.

Soe et al. [31] suggests a lightweight, sequential attack detection architecture that is based on 4 modules specifically to be used on IoT devices. The 4 modules are data collection, data categorisation, feature selection and model training. The first module data collection ingests benign and attack data from the network and IoT environment. Module 2 categorises each piece of attack and benign network data. Furthermore, each category includes all data with the same attack class and benign data. The feature selection module designates the highest correlated features of each class, as there are 8 types of attack class, the feature selection module will select features for these classes. The model selector module places and then evaluates several different ML algorithms, selecting the more accurate one. The attack detection occurs afterwards and involves feature extraction and then alert generation. A critique for this paper is that this algorithm claims to be lightweight, however, it is itself quite complex. This is evident as there are 4 modules required for this algorithm to operate, one of them being feature selection and as stated above is memory intensive. Moreover, the claim of lightweight is also contradicted by the operation of said algorithm. As the algorithm requires data collection, data categorisation, and feature selection to occur one after the other. After the feature selection occurs, the pre-processing of the attack detector begins, which then further involves attack detection and the subsequent alert generation. All this adds to the inherent complexity of the algorithm [31].

Samrin and Vasumathi [32] suggest an algorithm that is a combination of the KM clustering algorithm and an artificial neural network (ANN). This algorithm is broken down into the training phase and the testing phase. The training phase involves firstly completing the KM clustering method above. Once the clusters have been found, each output cluster from the KM clustering operation is trained by the ANN. This step keeps repeating for each neural network node till the output is produced, the training phase occurs afterwards. The training phase involves the total usage of the data, redistributed back through the KM algorithm, as well as the ANN in order to learn through itself. After the self-learning has taken place, the algorithm's accuracy, sensitivity, and specificity are returned. The accuracy scores were dependent on the number of clusters that were found within the data. Cluster sizes of 10, 15, 20, 25 and 30 had accuracy ratings of 88%, 89%, 92%, 88% and 89% respectively. The sensitivity (probability the algorithms predict positive examples) was found to be 80, 83, 76, 73 and 83%, once again depending on the cluster size. Lastly, the specificity rating (probability algorithms can predict negative examples) is rated as 66, 68, 68, 69 and 59%. It should be noted that the accuracy was better in every cluster, as well as the sensitivity. The specificity measurement proved to be negligible, with neither good nor bad results [32].

Saputra et al. [33] used a combination of NB and KM to aid with the classification of illiteracy. This is done in several steps, research data, pre-processing, clustering and classification, and lastly the testing method. The first step operates by gathering data from different sources, such as high and elementary schools, as well as ascertaining characteristic data such as unemployment rates and education enrolment percentage and illiteracy rate. After the research data step, pre-processing takes place and involves the combining of the accumulated data into 1, assessable table. The data is therefore pruned to only include related data, and then standardised so it is usable for mining purposes. The third step, clustering and classification, occurs next, with the KM being implemented, first forming 2, then 3, then 5 clusters in order to ascertain the different illiteracy levels. This step is completed with the goal of finding the number of clusters that are considered optimal and can then be used in the classification step with the NB algorithm. In the classification process, the NB was repeated 3 times using the training data, this will allow for the assessment of the types of illiteracy levels can be found. Lastly, the testing method involves utilising the k-fold method, in which 10 folds are used to validate results on the experimental data. After this is carried out, the results are analysed using accuracy and error rate as the metrics of analysis. The results for this indicate that the NB algorithm is a good candidate for the use of classifying clustered objects and finding anomalies. Furthermore, 3 clusters are ideal to be used. It should be noted that [33] did not use any feature selection algorithms, cutting down on operational time. They also got final accuracy ratings of 93% upwards, for each run of the algorithm. This suggests that using a combination of KM and NB gives strong results, this once again reinforces the proof of concept that KM and NB together enable a strong detection algorithm [33].

Varuna and Natesan [34] continue this same path with focusing on combining NB and KM together for an anomaly detection algorithm, this time specifically on the networking data involving the NSL-KDD set. This paper's algorithm is broken down into 3 stages, clustering, calculating the distance sum and classification. The first step involves clustering utilising the KM algorithm and grouping the objects into similar groups. Next, the original dataset is transformed into a newer dataset including the previously clustered samples. The newer dataset includes the combination of the training and testing data, which was used to calculate the K distance sums for each sample. The classification step involves the training dataset, being used to construct an NB classifier. There are several issues with this paper and its subsequent results. Varuna et al. discuss the metrics for evaluating the proposed algorithm, listing detection rate, false-positive rate, and accuracy. They then go on to not mention what these are for each step or give any other details regarding their results. The results that are provided indicate that this proposed algorithm scored lower in 2 out of 5 sections (the normal and dos predictions). Another issue that can be seen is that the way with this architectural framework is set out. The algorithm appears to be complicated in terms of operation, and the results not as strong. If something is used in AD, you would naturally expect stronger results the more steps that are involved, as this would indicate a more advanced thought process. This is not the case as mentioned above, the results would be considered average at best despite the intricate nature of the algorithm.

Tayal et al. [35] propose a way to detect spam in mail servers utilising a modified KM algorithm as well as NB for classification, this is outlined in the 7-step process. The first step involves the establishment of the mail server. Next, the dataset is collected and then placed into two sections, the training data and testing data with a 60% and 40% breakdown respectively. Step 3 involves the pre-processing of data by partitioning two parts, the header and body of the email. The header has general information such as sender id, date, time, subject and internet service provider. The body of the email contains the message of the sender. The pre-processing step is completed to assist the algorithm with interpreting the database and as such assisting the readability of the data. The pre-processing step consists of feature extraction, dimensionality reduction, evacuation of stop words and stemming. This allows for tokenisation to occur. Tokenisation refers to the idea of representing the broken-down words as tokens, allowing for the algorithm to read and subsequently operate computations on them. Step 4 involves term selection, and revolves around the frequency of the associated token, in every database, the frequency of said token demonstrates how often a word appears. Step 5 involves the operation of a modified KM algorithm that segregates the email body, into groups of similar messages based on the premise of comparability via the Euclidean distance. The modified KM eliminates the empty clusters expanding spam recognition. Step 6 relies on the NB classification and suggests based on probability whether the message is spam. Lastly, step 7 is the results step in which after computation occurs, the precision will be ascertained and displayed. Tayal et al. [35] state that this proposed algorithm returns rates of 96% precision in the detection of spam which is good. However, there are no specific mentions of what a base NB, KM or modified KM gives. It should be noted that there is a graph that shows accuracy rates and that the NB is depicted as having ~78%, the KM as ~90% and modified KM as ~91%. It is hard to see how the authors have rated each algorithm specifically, it is not stated within the text [35]. The research gaps found within the texts identified that although a similar algorithm to the proposed one has been used previously, it has not been used on IoT sensor data.

## 3 Proposed Algorithm

The Proposed algorithm has several requirements that need to be met, for it to be deemed a stronger alternative to the traditional AD algorithms. It should be noted that the below listed requirements are all needed to answer research question 1. The requirements are:

(1) Accuracy
(2) Speed
(3) Scalability
(4) Flexibility.

Each of these can also be used to answer the other listed questions. Good accuracy is required to answer research question 2. Fast speeds are required to answer research question 3. High scalability is required to answer research question 4. Strong flexibility is required to research question 5. The requirement of accuracy refers to the idea that the AD algorithm must be able to consistently predict the correct data. Speed refers to the training and testing times of the algorithm. Scalability means that the proposed algorithm must be able to be used on both small and large datasets. Flexibility means that it must be able to be used across different types of devices and to a high standard.

### *3.1 KMANB*

The KM Adaboosted Naive Bayes (KMANB) combines the strengths of the KM clustering algorithm, with the strength of the Naive Bayes SML algorithm. As mentioned above, the KM clustering algorithm is particularly strong at data aggregation, meaning it can produce clusters quickly. This works in conjunction with NB as its training times are quick, and testing scores are accurate. The clustering works with the sensor data, as this adds additional, soft clusters to the dataset, ensuring that there is more data to be classified by the NB. If there are correctly clustered samples within the set IE. A normal packet is correctly clustered with the other normal data, then NB would theoretically learn off these and be able to adequately predict the other anomalies within the IoT sensor data. The KMANB algorithm is depicted below.

The way in which the algorithm operates is depicted above in Fig. 1. Figure 1 shows that the KMANB algorithm can be broken down into 3 steps. Step 1 is the preparation phase, step 2 is the activation phase and step 3 is the evaluation phase.

### Step 1—Preparation

Step 1 involves the selection of the dataset, as well as any data manipulation and pre-processing. Pre-processing involved changing the labelling on the dataset, from a 1 or 0 to normal or anomaly. This was done to make it easier to understand. Next, the data was then normalised to ensure that the weights of different scales did not skew the data. It should also be noted that in some of the other datasets, another step was added to the pre-processing section to ensure that the clustering could be actioned. An example of this is with changing the string (Boolean) data of sphone_signal to nominal, in the IoT garage door train and test dataset. The use of Boolean was not allowing the clustering to occur.

### Step 2—Activation

The activation step refers to the initial operation of the KM clustering algorithm, the addition of that cluster to the set, and then the use of an Adaboosted Naive Bayes for the classification step. The activation of the KM algorithm includes specifying the number of clusters to be used, as well as any attributes to be ignored and the type of clustering to be performed and compared to a class of features. Firstly, the number of clusters that are specified relies on the different types of anomalies plus the one extra for normal. This can be represented as $C$ (clusters) $= A$ (anomalies) $+ 1$ (normal). An example is that the IoT Fridge has 6 anomalies and 1 normal, therefore this would be denoted as $C = 7$ or $C = 6 + 1$. Once the clusters have been set, the algorithm is run with a class to cluster evaluation, ignoring both type and label within the set. This ensures that the unsupervised element of this algorithm is maintained, and that the KM clustering looks at the data uninfluenced and then clusters accordingly. Lastly, the classification step happens when the Adaboosted Naive Bayes (ANB) is applied to the clustered data. The ANB uses the learnt clusters and then bases its predictions off that, meeting the supervised portion of the algorithm.

### Step 3—Evaluation

The third and final step is the evaluation portion of the algorithm. This refers to the algorithm being evaluated by looking at each of the clusters and their respective scores, then presenting an overall score. These scores will be discussed in a later section.

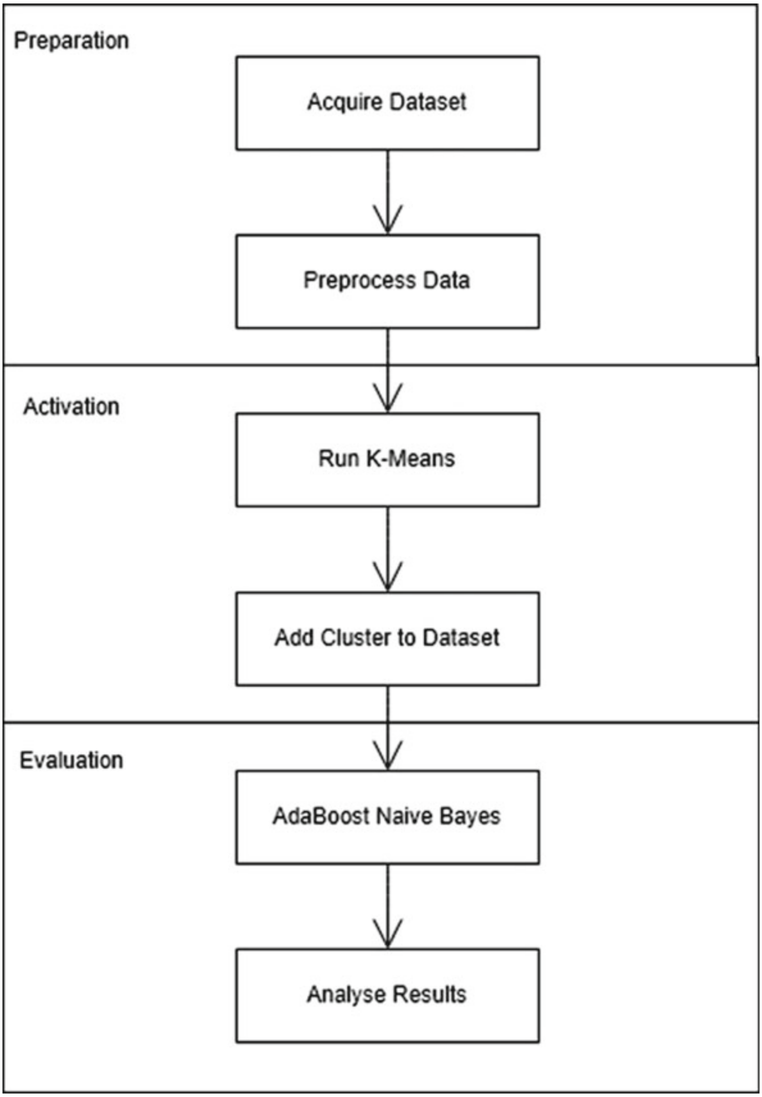


**Fig. 1** The 3 steps of the KMANB Algorithm

### *3.2 Algorithm Design*

The design of this algorithm is largely based on the idea of maximising strengths and minimising the weaknesses of each type of algorithm. Furthermore, it was decided against any feature selection techniques within the main body of the algorithm, due in large to the dataset being relatively simple. This goes against a lot of traditional thinking regarding anomaly detection, where feature selection is regarded as an integral step in dimensionality reduction [36]. In fact, we have done the opposite and have added more data to the set, this once again is in step with the trade-off with speed, precision and data size. It will be suggested that this addition has allowed for an increase in accuracy at the negligible cost of further clusters within the dataset.

## 4 Evaluation

The methodology presented below will demonstrate the actions being taken to address the need for an accurate HML algorithm to be used within IoT sensor data. This will include examining the different datasets. It should be noted that the datasets were provided by [11].

### *4.1 Methodology*

Our methodology involved firstly analysing and discussing many various texts within the academic sphere, to find a research gap. Once this gap was identified, 5 different datasets were researched, to assess which one would best suit an experiment regarding IoT smart devices and sensor data. The ToN_IoT Dataset was chosen. The ToN_IoT dataset is based on the new generation of IoT or IIoT (industrial internet of things) devices (IoT 4.0). ToN_IoT is used to evaluate different cyber security applications, as well as other artificial intelligence and, machine or deep learning algorithms. The ToN_IoT sets can be broken down into 4 different datasets; these are raw datasets, processed datasets, train and test datasets and security event ground truth datasets. The first KMANB algorithms were run on the base Train and Test datasets, to compare to the traditional SML algorithms and to subsequently examine whether the proposed algorithm was firstly accurate, and then to see if it maintained accuracy over the over devices. Once this was confirmed, correlation-based feature subset selection was run to discover the highest ranked feature to the anomaly label, and then remove it. Once removed, the experiments were run again to test whether the KMANB would maintain its accuracy without the strongest feature correlation related to anomaly type. After this, the processed (larger) datasets were manipulated and cut down slightly to make them operatable, whilst still maintaining its larger size and uneven distribution of anomalies. These experiments were then actioned to see if the KMANB was scalable to the larger datasets. These processed datasets were also run without the highest ranked feature relating to anomaly type, to further investigate the potential for the trade-off between speed and accuracy [11].

**Table 1** IoT train and test dataset statistics

| IoT device train and test dataset statistics | | | |
|---|---|---|---|
| Device | Rows | Columns | Size (KB) |
| Fridge | 59,945 | 6 | 2,617 |
| Garage door | 59,588 | 6 | 2,740 |
| GPS tracker | 58,961 | 6 | 3,422 |
| Modbus | 51,107 | 8 | 2,959 |
| Motion light | 59,489 | 6 | 2,416 |
| Thermostat | 52,775 | 6 | 2,547 |
| Weather sensor | 59,261 | 7 | 4,132 |

**Table 2** IoT train and test fridge feature descriptions

| IoT train and test fridge feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Fridge temperature | Number | Temperature measurement |
| 4 | Temp_condition | String | Temperature conditions of the sensor, State whether it is high or low |
| 5 | Label | Number | Normal or anomaly tag |
| 6 | Type | String | States the different types of attacks such as dos or backdoor |

## 4.2 Train and Test Dataset

The IoT Train Test dataset consists of 7 different IoT devices which are a fridge, garage door, GPS tracker, modbus, motion light, thermostat and weather sensor. Each of these datasets had different sizes and different data types. Table 1 provides an overview of each dataset size, including the number of rows, columns and size of the file. Tables 2, 3, 4, 5, 6, 7 and 8 give a profile of each IoT device. These device profiles contain the date, time, both labels (attack or normal and type of anomaly) as well as device-specific sensor information. Table 9 depicts the number of anomalies to be found within each IoT dataset.

## 4.3 Processed Dataset

The IoT processed datasets were used to test the concept of scalability. The results were then examined to see whether the algorithm maintained its accuracy when operating on larger datasets. The IoT processed datasets include more rows, file

**Table 3** IoT train and test garage door feature descriptions

| IoT train and test garage door feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Door_state | Boolean | State of the door sensor (true or false) |
| 4 | Sphone_signal | Boolean | State of the receiver of the door signal (true or false) |
| 5 | Label | Number | Normal or anomaly tag |
| 6 | Type | String | States the different types of attacks such as dos or backdoor |

**Table 4** IoT train and test GPS profile descriptions

| IoT train and test GPS feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Latitude | Number | Latitude value of GPS tracker |
| 4 | Longitude | Number | Longitude value of GPS tracker |
| 5 | Label | Number | Normal or anomaly tag |
| 6 | Type | String | States the different types of Attacks such as dos or backdoor |

**Table 5** IoT train and test modbus feature descriptions

| IoT train and test modbus feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | FC1_Read_Input_Register | Number | Modbus function that reads the input register |
| 4 | FC2_Read_Discrete_Value | Number | Modbus function that reads the discrete value |
| 5 | FC3_Read_Holding_Register | Number | Modbus function that reads the holding register |
| 6 | FC4_Read_Coil | Number | Modbus function that reads a coil |
| 7 | Label | Number | Normal or anomaly tag |
| 8 | Type | String | Different types of attacks such as dos or backdoor attacks |

**Table 6** IoT train and test motion light feature description

| IoT train and test motion light feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Motion_status | Number | Status of motion light (0 or 1) |
| 4 | Light_status | Boolean | Status of the light sensor (on or off) |
| 5 | Label | Number | Normal or anomaly tag |
| 6 | Type | String | Specifies types of attacks |

**Table 7** IoT train and test thermostat feature descriptions

| IoT train and test thermostat feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Current_temp | Number | Temperature reading |
| 4 | Thermostat_status | Boolean | Thermostat status (on or off) |
| 5 | Label | Number | Normal or anomaly tag |
| 6 | Type | String | Different types of attacks such as dos or backdoor attacks |

**Table 8** IoT train and test weather feature descriptions

| IoT train and test weather feature descriptions | | | |
|---|---|---|---|
| ID | Feature | Type | Description |
| 1 | Date | Date | Date of IoT logging |
| 2 | Time | Time | Time of logging IoT data |
| 3 | Temperature | Number | Temperature measurement from sensor |
| 4 | Pressure | Number | Pressure measurement from sensor |
| 5 | Humidity | Number | Humidity measurement from sensor |
| 6 | Label | Number | Normal or anomaly tag |
| 7 | Type | String | Different types of attacks such as dos or backdoor attacks |

sizes and different amounts of anomalies. Table 10 shows the different devices, as well as the rows, columns, and size of each file. Table 11 depicts the number of anomalies present within each dataset.

Both the train and test and the processed datasets were evaluated using WEKA (Waikato environment for knowledge analysis) version 3.8.4. This was done on a virtual machine within a Cyber Range, provided by Griffith University, School of

**Table 9** IoT train and test anomaly statistics

| | IoT train and test anomaly statistics | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Normal | Password | XSS | DDoS | Ransomware | Injection | Backdoor | Scanning |
| Fridge | 35,000 | 5000 | 2042 | 5000 | 2902 | 5000 | 5000 | 0 |
| Garage Door | 35,000 | 5000 | 1156 | 5000 | 2902 | 5000 | 5000 | 529 |
| GPS | 35,000 | 5000 | 577 | 5000 | 2833 | 5000 | 5000 | 550 |
| Modbus | 35,000 | 5000 | 577 | 0 | 0 | 5000 | 5000 | 529 |
| Motion Light | 35,000 | 5000 | 449 | 5000 | 2264 | 5000 | 5000 | 1775 |
| Thermostat | 35,000 | 5000 | 449 | 0 | 2264 | 5000 | 5000 | 61 |
| Weather | 35,000 | 5000 | 866 | 5000 | 2865 | 5000 | 5000 | 529 |

**Table 10** IoT processed dataset statistics

| IoT processed dataset statistics | | | |
|---|---|---|---|
| Device | Rows | Columns | Size (KB) |
| Fridge | 293,009 | 6 | 13,116 |
| Garage door | 89,754 | 6 | 4,286 |
| GPS tracker | 222,325 | 6 | 30,010 |
| Modbus | 198,173 | 8 | 12,013 |
| Motion light | 242,526 | 6 | 10,251 |
| Thermostat | 185,840 | 6 | 8,064 |
| Weather sensor | 251,045 | 7 | 16,331 |

**Table 11** IoT processed dataset anomaly statistics

| | IoT processed dataset type statistics | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Normal | Password | XSS | DDoS | Ransomware | Injection | Backdoor | Scanning |
| Fridge | 206,758 | 28,425 | 2042 | 10,233 | 2902 | 7079 | 35,568 | 0 |
| Garage door | 25,807 | 16,617 | 0 | 10,230 | 0 | 6331 | 30,230 | 529 |
| GPS | 140,488 | 25,176 | 577 | 10,226 | 2833 | 6904 | 35,571 | 550 |
| Modbus | 133,839 | 18,815 | 498 | 0 | 0 | 5186 | 40,005 | 529 |
| Motion light | 178,591 | 17,521 | 449 | 8121 | 2264 | 5595 | 28,209 | 1775 |
| Thermostat | 129,563 | 8435 | 449 | 0 | 2264 | 9498 | 35,568 | 61 |
| Weather | 160,529 | 25,715 | 866 | 15,182 | 2865 | 9726 | 35,641 | 529 |

Information and Communication Technology. WEKA was chosen because of its ease of use and its ability to utilise different algorithms.

## 4.4 Hypothesis

The hypothesis for this experiment suggests that the combination of UML and SML, creating a HML algorithm, will be of comparable, if not better strength compared to the traditional IoT ML algorithms. The proposed KMANB will be more accurate, as well as having a similar if not quicker training and testing time than the more traditional SML algorithms. Furthermore, it is hypothesised that the KMANB will be more flexible compared to the other SML algorithms as well as being scalable in nature. The algorithm will be assessed by looking at the accuracy, precision and recall scores (APR scores), as well the training and testing times (speed). Although the ideas of flexibility and scalability are not easily quantified, these requirements will be assessed by looking at the overall APR scores over different devices and the larger dataset APR scores respectively. The APR scores are defined below.

## 4.5 Evaluation Metrics

WEKA's experiment function allows for the generation of true positive (TP), true negative (TN), false positive (FP) and false-negative (FN) numbers, which are then used to calculate the APR scores. TP refers to the number of correctly predicted data points, TN are predicted false and turn out to be false. FP are flagged as positive, with their true value being negative, and FN are predicted negatives, that are in fact true. The accuracy was calculated by adding the total number of true positives and true negatives together and then dividing them by the total of the true positives, true negatives, false positives and the false negatives. Precision involves true positives divided by the sum of the true positives and false positives. The recall involves dividing true positives by the sum of true positives and false negatives. These equations can be seen below [37].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

After the experiments were run, the number of TP, TN, FP, FN is then presented by WEKA. These will then be placed into a spreadsheet with the accuracy (Acc), precision (Pre) and recall (Rec) formulas. The spreadsheet then presented the calculated APR scores. It should also be noted that CFS was used to evaluate both the train and test datasets as well as the larger processed datasets. This was done to find the highest ranked correlation to anomaly type, and aid in its removal to test the feature reduction research question. The suitability for these metrics refers to the idea of what makes a strong, AD system.

## 5 Results

The results of the KMANB HML algorithm will be compared to 3 other SML algorithms using the Train and Test dataset as a baseline. This will be done to demonstrate proof of concept and to examine the performance of our algorithm compared to traditional SML methods.

### *5.1 Train and Test Results*

The below tables in this section are three SML algorithms being compared against the proposed KMANB algorithm.

### *5.2 Train and Test with no Highest Ranked Feature*

The below results are the Train and Test datasets with the feature Date removed. CFS was run to find the feature with the highest correlation to anomaly type and remove it. This was done to compare the training and testing times and investigate whether data reduction decreases the train and test time, as well as negatively impacting the APR scores. If this algorithm produces strong results even with the removal of the highest-ranked correlation, it will further solidify the idea that it is a reasonable alternative to the traditional SML algorithms. It should also be noted that KMANB is the only algorithm being tested to see if feature reduction improves the training and testing times whilst maintaining the APR scores. As such, it was not necessary to include the traditional SML algorithms times in the below Figs.

### *5.3 Processed Dataset Results*

The train and test datasets were used as a proof of concept. The processed datasets results are examined to test the scalability of the KMANB, and to ascertain if it could be applicable in real world situations on larger, uneven datasets.

### *5.4 Processed Dataset no Highest Ranked Feature*

The processed dataset without the highest ranked feature results is recorded below. This was done to ascertain whether the highest ranked correlation to the anomaly type, would affect the APR results in the larger datasets, the same as the above smaller datasets.

## 6 Discussion

The results from the KMANB algorithm experiments will be described looking at the following key criteria: accuracy, speed, scalability and flexibility, as well as taking into consideration the associated APR scores. The experiments will also answer the research questions posed by this chapter.

### *6.1 Accuracy, Precision and Recall Scores*

Firstly, the KMANB is overall stronger regarding accuracy, precision and recall within the different IoT systems. This therefore fulfills the flexibility question as it has scored well across all devices as well as being accurate overall.

The KMANB scored higher APR scores than RF, NB and KNN. This was seen as Table 12, shows KMANB scored with a 0.99 on all 3 of APR, as opposed to the RF algorithm, which scored a 0.97 on all 3 measures, and the NB which scored 0.53 on the accuracy, precision, and a 0.51 on the recall aspect. The other section of Table 12 showed that all tested algorithms gave APR scores as 1. The Table 13 experiments found that once again, the KMANB was the highest rated across all three APR measurements, with the scores being presented at 0.99, 0.99, and 0.95 respectively. This is compared to the RF scores of 0.85, 0.85, 0.85, NB with 0.84, 0.86, 0.85, and KNN scores of 0.88, 0.89, 088 respectively. Furthermore, the Modbus scores of Table 13 also show the strength of KMANB, with it being rated as 0.98, 0.95 and 0.96 APR scores respectively. This is compared to the 0.77 for KNN, 0.67 for NB and RF measured at 0.97. However, it should be noted that the RF scored 0.98 for precision and recall, as opposed to 0.95 and 0.96 of the KMANB. KMANB

**Table 12** IoT train and test fridge and garage door experiment results

| IoT train and test fridge and garage door experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fridge | | | | Garage door | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.97 | 0.53 | 0.99 | 0.99 | 1 | 1 | 1 | 1 |
| Pre | 0.97 | 0.53 | 0.99 | 0.99 | 1 | 1 | 1 | 1 |
| Rec | 0.97 | 0.51 | 0.99 | 0.99 | 1 | 1 | 1 | 1 |
| Train | 0.188 | 0.011 | 0.147 | 2.98 | 0.062 | 0.010 | 0.625 | 3.38 |
| Test | 0.045 | 0.005 | 2.556 | 0.44 | 0.000 | 0.002 | 0.969 | 0.48 |

**Table 13** IoT train and test GPS and modbus experiment results

| IoT train and test GPS and modbus experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | GPS | | | | Modbus | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.85 | 0.84 | 0.88 | 0.99 | 0.97 | 0.67 | 0.77 | 0.98 |
| Pre | 0.85 | 0.86 | 0.89 | 0.99 | 0.98 | 0.46 | 0.77 | 0.95 |
| Rec | 0.85 | 0.85 | 0.88 | 0.95 | 0.98 | 0.68 | 0.78 | 0.96 |
| Train | 0.833 | 0.009 | 0.08 | 5.39 | 1.587 | 0.012 | 0.060 | 4.97 |
| Test | 0.099 | 0.007 | 1.508 | 0.78 | 0.031 | 0.002 | 0.116 | 0.70 |

scored second-highest overall, as compared to NB and KNN but less than the standard RF algorithm. Table 14 showed that KMANB is accurate for the motion light IoT device. The APR score was 1,1 and 1 respectively. Compared to the RF and its rating of 0.58, 0.34, and 0.59, NB with 0.66, 0.44 and 0.66 as well as KNN with 0.60, 0.56, and 0.61. The thermostat section of Table 14 shows ratings of 0.99, 0.97 and 0.93 respectively for the KMANB. This contrasts with RF and its 0.66, 0.55 and 0.66, NB and 0.66, 0.44 and 0.66, and lastly KNN and 0.60, 0.56 and 0.61. Table 14 also reinforced the idea of the KMANB's accuracy, with ratings of 0.99, 0.97 and 0.93 respectively. Once again, this is compared to the scores for RF (0.66, 0.55 and

**Table 14** IoT train and test motion light and thermostat experiment results

| IoT train and test motion light and thermostat experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Motion light | | | | Thermostat | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.58 | 0.58 | 0.54 | 1 | 0.66 | 0.66 | 0.60 | 0.99 |
| Pre | 0.34 | 0.34 | 0.34 | 1 | 0.55 | 0.44 | 0.56 | 0.97 |
| Rec | 0.59 | 0.59 | 0.59 | 1 | 0.66 | 0.66 | 0.61 | 0.93 |
| Train | 0.1 | 0.011 | 3.157 | 2.44 | 1.044 | 0.009 | 0.064 | 4.13 |
| Test | 0.008 | 0.002 | 6.409 | 0.34 | 0.023 | 0.002 | 0.088 | 0.62 |

0.66), NB (0.66, 0.44 and 0.66) and KNN (0.60, 0.56 and 0.61). There were several interesting results in terms of APR that have been found. One result was that Table 12 garage door section scores were all found to be 1. This could be because of several different reasons, although [11] suggests that it's because of the type of data that is found within the dataset. The data is of a discrete nature, meaning that it only has a certain number of values to be counted, an example of this is the number of students in a classroom. This means that each algorithm might have been able to adequately predict each outcome as the data.

These above scores demonstrate the overall strength of the KMANB, and show that compared to traditional SML algorithms, it is just as, if not a stronger choice. The accuracy of the KMANB could be due to the pre-classification step of the KM algorithm. As the KM algorithm was used to generate a new cluster within the dataset, the NB would have theoretically seen that cluster, ingested, and trained from it. After the training took place, the testing step would have then utilised what was learnt from the already clustered classes within the dataset, and then based its predictions on that. This is also reinforced by a strength of HML, as UML could theoretically help SML with providing more data points to base its labels from.

## 6.2 Train and Test Times

The speed (train and test time) for the KMANB was not as competitive as first thought. Looking at Table 12, the KMANB test on the fridge came in at 2.98 s and then 0.44 for a total time of operation for 3.42 s. This is far behind the time of RF, NB and KNN with total times of 0.233, 0.115 and 2.703 respectively. The scores for the garage door in Table 12 are logged as 3.38 train and 0.48 test time, meaning a total time of 3.86 was recorded. This once again in comparison to the scores of the RF, NB and KNN of 0.062, 0.012 and 1.594 respectively. The Table 13 for the GPS tracker KMANB speed times are a total of 6.17 total time (5.39 train and 0.78 test). This is contrasted to the scores of 0.932, 0.016 and 0.08 for the 3 SML algorithms. The results for the Modbus (Table 13) suggested that once again, our proposed algorithm operated several seconds behind, with a training time of 4.97 and a testing time of 0.70 giving a total operation time of 5.67. The scores for the RF, NB and KNN were found to be in totality, 1.618, 0.014 and 0.176. The Motion light (Table 14) results showed that KMANB was slower with a total time of operation for 2.78. The RF and NB times were found to be quicker, with total times of 0.108, 0.01. However, it should be noted that the KNN time was far slower with a total operation time of 9.566, compared to the proposed algorithm. Table 14 thermostat found the KMANB was slower once again. The RF, NB and KNN algorithms were found to be run at 1.067, 0.011 and 0.152 respectively while the KMAND spent a total of 4.75 s in operation. Lastly, Table 15 showed that the KMANB spent 7.2 s in operation, versus the RF, NB and KNN scores of 0.797, 0.013 and 0.48 s of operation.

It should be noted that Table 14 motion light scores for KMANB were slower than the RF and NB scores, but faster than the KNN scores. This could be for

**Table 15** IoT train and test weather experiment results

| | IoT train and test weather experiment results | | | |
|---|---|---|---|---|
| | Weather | | | |
| | RF | NB | KNN | KMANB |
| Acc | 0.84 | 0.69 | 0.81 | 0.97 |
| Pre | 0.84 | 0.72 | 0.81 | 0.87 |
| Rec | 0.84 | 0.69 | 0.81 | 0.90 |
| Train | 0.789 | 0.011 | 0.066 | 6.28 |
| Test | 0.008 | 0.002 | 0.414 | 0.92 |

several reasons. As KNN assumes that everything that is close, is related, it would try to classify data that is already grouped together, however light_status as shown in Table 6, is Boolean. Meaning its either True or False (on or off), as such this might have affected the accuracy of class membership designation. Furthermore, the way in which KNN operates, requires the total ingestion of the data first before calculations can occur. This means that the algorithm would have had to essentially load everything up first. This is contrasted to NB that only works on assumptions of independence, not requiring the entire dataset at once to be ingested.

The slower times of the KMANB could be affected by the hardware the test was run on as this can either speed up or slow down the operation. As our tests were run on a VM with 2 CPU's ~2.7 GHZ, our scores could have been negatively impacted by this. However, this is not suggesting that KMANB is quicker, it does in fact appear to be generally slower, but just not as slow as is being shown. These results suggest that the proposed algorithm is not as fast comparatively to the traditional SML algorithms. This leads to the idea of examining what is the correct amount of data reduction needed to find the balance between speed and precision of an AD algorithm. Too little accuracy and the algorithm is useless, too little speed and it becomes redundant. The following No Highest Ranked Feature dataset experiments were run to ascertain if our algorithms speed could be improved whilst maintaining its APR scores.

## 6.3 Algorithm Optimisation

Firstly, looking at Table 16, the training and testing times of the KMANB fridge experiment were still high comparatively, coming in at 2.88 and 0.44. The garage door (Table 16) yielded 0.92 training and 0.12 testing times. The Table 17 GPS experiment gave 4.39 training and 0.66 testing time. The Modbus training and testing times produced 4.43 and 0.64 times respectively. Table 18 motion light was found to be 1.63 training and a 0.22 testing time. Thermostat (Table 18) was found to have times of 3.49 and 0.54 respectively train and test times respectively. Lastly, Table 19 weather sensor registered a times of 5.24 and a 0.79.

**Table 16** IoT fridge and garage door no highest ranked experiment results

| IoT fridge and garage door no highest ranked experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fridge | | | | Garage door | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.97 | 0.53 | 0.99 | 0.99 | 1 | 1 | 1 | 0.99 |
| Pre | 0.97 | 0.53 | 0.99 | 0.99 | 1 | 1 | 1 | 0.99 |
| Rec | 0.97 | 0.51 | 0.99 | 0.99 | 1 | 1 | 1 | 0.99 |
| Train | | | | 2.88 | | | | 0.92 |
| Test | | | | 0.44 | | | | 0.12 |

**Table 17** IoT GPS and modbus no highest ranked experiment results

| IoT GPS and modbus no highest ranked experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | GPS | | | | Modbus | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.85 | 0.84 | 0.88 | 0.99 | 0.97 | 0.67 | 0.77 | 0.98 |
| Pre | 0.85 | 0.86 | 0.89 | 0.94 | 0.98 | 0.46 | 0.77 | 0.96 |
| Rec | 0.85 | 0.85 | 0.88 | 0.96 | 0.98 | 0.68 | 0.78 | 0.95 |
| Train | | | | 4.39 | | | | 4.43 |
| Test | | | | 0.66 | | | | 0.64 |

**Table 18** IoT train and test motion light and thermostat experiment results

| IoT train and test motion light and thermostat experiment results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Motion light | | | | Thermostat | | | |
| | RF | NB | KNN | KMANB | RF | NB | KNN | KMANB |
| Acc | 0.58 | 0.58 | 0.54 | 0.99 | 0.66 | 0.66 | 0.60 | 0.99 |
| Pre | 0.34 | 0.34 | 0.34 | 0.99 | 0.55 | 0.44 | 0.56 | 0.97 |
| Rec | 0.59 | 0.59 | 0.59 | 1 | 0.66 | 0.66 | 0.61 | 0.91 |
| Train | | | | 1.63 | | | | 3.49 |
| Test | | | | 0.22 | | | | 0.54 |

**Table 19** IoT train and test weather with no highest ranked experiment results

| IoT train and test weather with no highest ranked experiment results | | | | |
|---|---|---|---|---|
| Weather sensor | | | | |
| | RF | NB | KNN | KMANB |
| Acc | 0.84 | 0.69 | 0.81 | 0.97 |
| Pre | 0.84 | 0.72 | 0.81 | 0.89 |
| Rec | 0.84 | 0.69 | 0.81 | 0.91 |
| Train | | | | 5.24 |
| Test | | | | 0.79 |

Examining the Table 16 fridge further, KMANB is ranked at 2.88 and 0.44 train and test speed, compared to Table 12 fridge which was 2.98 and 0.44. This was due in large to the highest correlation feature of date being removed and it not affecting the score. This means that the reduction of data leads to KMANB being able to operate quicker. Another experiment that can be examined, would be Table 19, which was the Weather Train and Test without the highest ranked feature, its scores are 5.24 and 0.79 for speed, as well as 0.97, 0.89 and 0.91 for APR, respectively. Compared to Table 15, with the scores of 6.28 and 0.92 for speed, and 0.97, 0.87 and 0.90 APR scores, respectively. The drop in the speed, and the increase in the precision and recall for Table 19 weather could be attributed to the highest feature correlation having too much weight and therefore influencing the algorithm and obfuscating some of the rankings. This could be because date is nominal, and one nominal attribute is counted as 1 in WEKA, if there are 100,000 data points on 1 day, that is 100,000 added to the weight of that day. This means that more numbers and a greater distribution of weighting is allocated to this feature. When date is removed, memory is freed up and the subsequent weight given to this attribute is removed.

Overall, the results suggest that accuracy was not sacrificed to the point where the algorithm becomes unreliable. An example of this is the garage door IoT (Table 16) dropping by 0.01–0.99 for all the APR measurements, whilst the train and test speed improving from 3.68 and 0.18, training and testing to 2.88 and 0.44. It should be noted that overall, the total time in operation is lowered by 0.54 s. The testing time increases marginally, this is due in large to the algorithm having less data to base its predictions off, but enough that it is not thrown out considerably. The trade-off for speed and accuracy stems from the idea that feature reduction is needed to ensure a faster, more precise algorithm when regarding anomaly detection within networks. It could be suggested that this is not specifically needed for IoT sensor data, as it is already less complicated in nature compared to the traditional TCP/IP data found within networking. Another result which reinforces the idea of feature of reduction involves Table 18. The no highest results demonstrated that although the accuracy was found to be the same, the precision was slightly more with a 0.02 increase but a decrease of 0.01 regarding the recall score. The time in training was found to be a total of 1.17 s slower. This demonstrates that within IoT sensor data, the trade-off can be achieved with minimal downside, further improving the algorithm and its chances of adoption however, it is not mandatory. Arguably, the most important facet of an AD algorithm is accuracy, and the ability to consistently produce strong results.

It should also be noted that the original plans for this algorithm involved adding an additional layer of KM clustering to go to 2 total clusters, clustered to normal and anomaly. The algorithm would then be further clustered to the formula of $C = A + 1$. This was done in order to see if adding an additional cluster assist with the final step and to produce higher results. It was found that it did not help, and as such it was decided not to be included within the final algorithm. This once again forced the examination of the trade-off between speed and accuracy. It was assessed as adding another layer of algorithm, increasing complexity, with no real gain.

## 6.4 Scalability of KMANB

The scalability of the KMANB was also examined. Scalability refers to the idea that once this has been applied to the smaller, even datasets, this concept is then taken and applied to larger, uneven datasets. Table 20 depicts all the results for the processed datasets. The fridge scores 0.99 for all 3 types of ratings, as does the garage door. The GPS tracker rates at 0.98, 0.92 and 0.82 for APR respectively. The Modbus results indicate a 0.98, 0.96 and 0.95 APR score. The motion light was found to have results of 0.99 for accuracy and precision, with a recall ranking of 0.98. The KMANB on thermostat was given an APR score of 0.98, 0.95 and 0.95, with the weather dataset's run resulting in a 0.98, 0.89 and 0.92 in scores. Looking at this, it can be suggested this algorithm is indeed scalable and even resulted in comparable readings in the bigger datasets, when compared to the smaller datasets. An example of this is the fridge's APR results were 0.99 for all the areas, the same as the larger datasets. Furthermore, the garage door was slightly less on the processed side, with 0.99 for all 3, compared to the smaller datasets with full 1s. Although this is indeed less, it still doesn't dissuade from the notion that this algorithm is indeed scalable, as a loss of 0.01 for all 3 sections still points to overall a strong algorithm. If the smaller weather dataset's results are examined, this one does indicate the larger dataset produced stronger results. The APR results for the smaller dataset were 0.97, 0.87 and 0.90, compared to 0.98, 0.89 and 0.92 respectively. This further demonstrates the scalable nature of this algorithm, as it is even, if not higher rated for the use of IoT anomaly detection with both smaller and larger datasets. Table 21 results also reiterated the trade-off between accuracy and speed. It depicts the APR scores of the fridge, garage door, GPS tracker, Modbus, motion light, thermostat, and weather sensor. It shows that

**Table 20** IoT processed dataset experiment results

| IoT processed dataset experiment results | | | | | | | |
|---|---|---|---|---|---|---|---|
| KMANB results | | | | | | | |
| | Fridge | Garage door | GPS tracker | Modbus | Motion light | Thermostat | Weather |
| Acc | 0.99 | 0.99 | 0.98 | 0.98 | 0.99 | 0.98 | 0.98 |
| Pre | 0.99 | 0.99 | 0.92 | 0.96 | 0.99 | 0.95 | 0.89 |
| Rec | 0.99 | 0.99 | 0.82 | 0.95 | 0.98 | 0.95 | 0.92 |

**Table 21** IoT processed dataset with no highest ranked experiment results

| IoT processed dataset with no highest ranked experiment results | | | | | | | |
|---|---|---|---|---|---|---|---|
| KMANB results | | | | | | | |
| | Fridge | Garage door | GPS tracker | Modbus | Motion light | Thermostat | Weather |
| Acc | 0.99 | 0.99 | 0.99 | 0.97 | 0.99 | 0.99 | 0.98 |
| Pre | 0.97 | 0.99 | 0.92 | 0.89 | 0.99 | 0.98 | 0.93 |
| Rec | 0.98 | 1 | 0.94 | 0.93 | 0.99 | 0.98 | 0.94 |

KMANB still maintained strength even with the removal of the highest correlation to anomaly type.

### *6.5 Traditional IoT AD Algorithms Versus the KMANB*

The KMANB algorithm appears to be a stronger choice for all the IoT devices listed above. The main reason why due to its higher APR scores, which indicates a stronger algorithm when it comes to AD. Furthermore, it appears to be flexible, maintaining its high scores across multiple devices and dataset sizes. This means that the research question “Is the proposed algorithm able to be applied to different types of IoT devices?” Is sufficiently answered. Although the KMANB is slower, it consistently provides the same APR scores across different devices, both including the smaller and larger datasets. Thus, the requirements of accuracy, flexibility and scalability were met.

## 7 Conclusion and Future Work

The research questions found within this chapter aim to discuss several facets of the proposed KMANB algorithm and its use on IoT sensor data. The first of these questions is arguably the most important one, and that is whether the KMANB is the stronger choice compared to traditional SML algorithms. Once this question has been answered, this then allows the further examination of sub theories.

One sub theory relates to whether the KMANB is faster in terms of total time in operation compared to the other SML algorithms. This was done by first utilising the train and test IoT sensor datasets and comparing the APR results for our data, against traditional SML algorithms. Next, the scalability of the proposed algorithm was discussed as well, with the KMANB needing to be used to test large slices of data to be found viable. This was achieved by using the processed datasets and seeing if our proposed algorithm was still accurate. Finally, the trade-off between speed and accuracy of the KMANB was also tested. This was done by comparing the two smaller datasets with one having the largest feature correlation relating the type of anomaly removed. The speed and APR measures were then compared in order to ascertain whether or not the score reductions were drastic and algorithm breaking. The research questions were all met, as our proposed algorithm rated higher in almost all the APR tests overall, maintained its high scores on the larger, uneven datasets. However, it should be noted that its speed was slower in some instances. This speed issue was addressed also, with the reduction of the highest feature correlation of an anomaly occurring. This resulted in the lowering of the time in operation on some devices.

This algorithm and its application to IoT sensor data has multiple avenues to be further examined. One such potential future work involves the isolation of the

IoT anomalies. As we have successfully identified them, further research could be completed in which some form of Principal Component Analysis is carried out on the data that is found to be anomalous. This could be done by exporting the correctly predicted data into a.csv file, and then loaded into WEKA. After this has occurred, PCA could be run, and then this could potentially be able to show what sensor information is aligned to the correctly predicted anomalies. Other future work that could be carried out regards the increase in scope for the KMANB and its use. As mentioned above, there are millions of different IoT devices, and as such the scope of this could be further increased to test specific brands of garage doors, fridges or GPS trackers. This would theoretically allow us to further examine the potential real-world applications for the proposed algorithm. Alternate reduction of different data could also be further examined. As the highest correlation to the type of anomaly was reduced in the no highest ranked feature datasets, further reduction of the biggest in terms of size could be done, as this would potentially not reduce the accuracy by a considerable amount and also help to reduce the time in operation.

## References

1. C. Wang, IoT anomaly detection method in intelligent manufacturing industry based on trusted evaluation. Int. J. Adv. Manuf. Technol. **107**, 993–1005 (2020). https://doi.org/10.1007/s00170-019-04274-0
2. P. Sethi, S. Sarangi, Internet of things: architectures, protocols, and applications. J. Electr. Comput. Eng. **1–25** (2017). https://doi.org/10.1155/2017/9324035
3. A. Khamparia, S. Pande, D. Gupta, A. Khanna, A. Sangaiah, Multi-level framework for anomaly detection in social networking. Library Hi Tech **38**, 350–366 (2020). https://doi.org/10.1108/LHT-01-2019-0023
4. D. Rawat, S. Reddy, Software defined networking architecture, security and energy efficiency: a survey. IEEE Commun. Surv. Tutor. **19**, 325–346 (2017). https://doi.org/10.1109/COMST.2016.2618874
5. P.I. Radoglou Grammatikis, P.G. Sarigiannidis, I.D. Moscholios, Securing the internet of things: challenges, threats and solutions. Internet of Things **5**, 41–70 (2019). https://doi.org/10.1016/j.iot.2018.11.003
6. C. Patel, N. Doshi, A novel MQTT security framework in generic IoT model. Procedia Comput. Sci. **171**, 1399–1408 (2020). https://doi.org/10.1016/j.procs.2020.04.150
7. F. Hussain, R. Hussain, S. Hassan, E. Hossain, Machine learning in IoT security: Current solutions and future challenges. IEEE Commun. Surv. Tutor. **22**, 1686–1721 (2020). https://doi.org/10.1109/COMST.2020.2986444
8. M. Usama, J. Qadir, A. Raza, H. Arif, K. Yau, Y. Elkhatib, et al., Unsupervised machine learning for networking: techniques, applications and research challenges. IEEE Access **7**, 65579–65615 (2019). https://doi.org/10.1109/ACCESS.2019.2916648
9. E. van Engelen Jesper, H.H. Hoos, A survey on semi-supervised learning. Mach. Learn. **109**(2), 373–440 (2020). http://dx.doi.org.libraryproxy.griffith.edu.au/https://doi.org/10.1007/s10994-019-05855-6
10. W. Kassab, K. Darabkh, A–Z survey of internet of things: architectures, protocols, applications, recent advances, future directions and recommendations. J. Netw. Comput. Appl. **163** (2020). https://doi.org/10.1016/j.jnca.2020.102663
11. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access **8**, 165130–165150 (2020). https://doi.org/10.1109/ACCESS.2020.3022862

12. Y. Quek, W. Woo, L. Thillainathan, IoT load classification and anomaly warning in ELV DC picogrids using hierarchical extended-nearest neighbors. IEEE Internet Things J. **7**, 863–873 (2020). https://doi.org/10.1109/JIOT.2019.294542566
13. N. Sahu, I. Mukherjee, Machine learning based anomaly detection for IoT network: (anomaly detection in IoT network), in *Machine Learning Based Anomaly Detection for IoT Network: (Anomaly Detection in IoT Network)* (2020). https://doi.org/10.1109/ICOEI48184.2020.9142921
14. C. Tsai, Y. Hsu, C. Lin, W. Lin, Intrusion detection by machine learning: a review. Expert Syst. Appl. **36**, 11994–12000 (2009). https://doi.org/10.1016/j.eswa.2009.05.029
15. M. Lawal, R. Shaikh, S. Hassan, An anomaly mitigation framework for iot using fog computing. Electronics (Basel) **9**, 1–24 (2020). https://doi.org/10.3390/electronics9101565
16. Y. Bengio, I.J. Goodfellow, A. Courville, Deep learning, book in preparation for mit press (2015). Disponıvel em http://www.iro.umontreal.ca/bengioy/dlbook
17. M. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A. Sheth, Machine learning for internet of things data analysis: a survey. Digital Commun. Networks, **4**, 161–175 (2018) https://doi.org/10.1016/j.dcan.2017.10.002
18. H. Zhang, Exploring conditions for the optimality of naive Bayes. Int. J. Pattern Recognit Artif Intell. **19**(02), 183–198 (2005)
19. Y. Wang, S. Xia, Q. Tang, J. Wu, X. Zhu, A novel consistent random forest framework: Bernoulli random forests. IEEE Trans. Neural Netw. Learn. Syst. **29**, 3510–3523 (2018). https://doi.org/10.1109/TNNLS.2017.272977868
20. H. Frigui, Unsupervised learning of arbitrarily shaped clusters using ensembles of gaussian models. Pattern Anal. Appl.: PAA **8**, 32–49 (2005). https://doi.org/10.1007/s10044-005-0240-y
21. W.L. Zhao, C.H. Deng, C.W. Ngo, k-means: a revisit, Neurocomputing, **291**, 195–206 (2018). ISSN 0925–2312, https://doi.org/10.1016/j.neucom.2018.02.072.
22. J. Qi, Y. Yu, L. Wang, J. Liu, Y. Wang, An effective and efficient hierarchical K-means clustering algorithm. Int. J. Distrib. Sens. Netw. **13**, 1–17 (2017). https://doi.org/10.1177/1550147717728627
23. N. Li, A. Martin, R. Estival, Combination of supervised learning and unsupervised learning based on object association for land cover classification, in *Combination of Supervised Learning and Unsupervised Learning Based on Object Association for Land Cover Classification* (2018). https://doi.org/10.1109/DICTA.2018.8615871
24. R. Kristianto, B. Santoso, M. Sari, (2019). Integration of K-means clustering and naïve bayes classification algorithms for smart AC monitoring and control in WSAN, in *Integration of K-means clustering and naïve bayes classification algorithms for smart AC monitoring and control in WSAN*. https://doi.org/10.1109/ICITISEE48480.2019.900392765
25. M. Wayahdi, Tulus, M. Lydia, Combination of k-means with naïve bayes classifier in the process of image classification. IOP Conf. Ser. Mater. Sci. Eng. **725**, 12126 (2020). https://doi.org/10.1088/1757-899X/725/1/012126
26. A. Allahverdipour, F. Soleimanian Gharehchopogh, A new hybrid model of k-means and naïve bayes algorithms for feature selection in text documents categorization. J. Adv. Comp. Res. **8**,73–86 (2017)
27. Z. Fadhil, Hybrid of K-means clustering and naive Bayes classifier for predicting performance of an employee. Period. Eng. Nat. Sci. (PEN) **9**(799–807), 64 (2021)
28. S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi, A. Pinto, Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset. Secur. Priv. **2**. https://doi.org/10.1002/spy2.91
29. P. Bhatt, B. Thakker, Mass removal of botnet attacks using heterogeneous ensemble stacking PROSIMA classifier in IoT. Int. J. Commun. Netw. Inform. Secur. **11**, 380–390 (2019)
30. H. Om, A. Kundu, A hybrid system for reducing the false alarm rate of anomaly intrusion detection system, in *A Hybrid System for Reducing the False Alarm Rate of Anomaly Intrusion Detection System* (2012). https://doi.org/10.1109/RAIT.2012.6194493
31. Y. Soe, Y. Feng, P. Santosa, R. Hartanto, K. Sakurai, Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors **4372** (2020). https://doi.org/10.3390/s2016437267

32. R. Samrin, D. Vasumathi, Hybrid weighted K-means clustering and artificial neural network for an anomaly-based network intrusion detection system. J. Intell. Syst. **27**, 135–147 (2018). https://doi.org/10.1515/jisys-2016-0105
33. M. Saputra, T. Widiyaningtyas, A. Wibawa, Illiteracy classification using K means-naïve bayes algorithm. JOIV: Int. J. Inf. Vis. **2**, 153–158 (2018). https://doi.org/10.30630/joiv.2.3.129
34. S. Varuna, P. Natesan, An integration of k-means clustering and naïve bayes classifier for intrusion detection, in *An Integration of k-Means Clustering and Naïve Bayes Classifier for Intrusion Detection* (2015). https://doi.org/10.1109/ICSCN.2015.7219835
35. D. Tayal, A. Jain, K. Meena, Development of anti-spam technique using modified K-means & naive bayes algorithm, in *Development of Anti-Spam Technique using Modified K-Means & Naive Bayes Algorithm* (2016), pp. 2593–2597
36. H.Y. Teh, K.I. Wang, A.W. Kempa-Liehr, Expect the unexpected: unsupervised feature selection for automated sensor anomaly detection. IEEE Sens. J. 1–1. https://doi.org/10.1109/JSEN.2021.3084970
37. M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations. Int. J. data min. knowledge manage. process, **5**(2), 1 (2015)
38. A. Colakovic, M. Hadzialic, Internet of things (IoT): a review of enabling technologies, challenges, and open research issues. J. King Saud Univ. Comput. Inform. Sci. **144**, 291–319 (2018). https://doi.org/10.1016/j.comnet.2018.07.017Ray
39. H. Jagadish, B. Ooi, K. Tan, C. Yu, R. Zhang, iDistance: an adaptive B+ tree based indexing method for nearest neighbor search. ACM Trans. Database Syst. **30**(2), 364–397 (2005). https://doi.org/10.1145/1071610.1071612
40. R. Memon, J. Li, M. Nazeer, A. Khan, J. Ahmed, DualFog-IoT: additional fog layer for solving blockchain integration problem in internet of things. IEEE Access **7**, 169073–169093 (2019). https://doi.org/10.1109/ACCESS.2019.2952472
41. S. Sharma, P. Pandey, S. Tiwari, M. Sisodia, An improved network intrusion detection technique based on k-means clustering via naïve bayes classification, in *An Improved Network Intrusion Detection Technique Based on k-Means Clustering Via Naïve Bayes Classification* (2012), pp. 417–422
42. S. Uddin, A. Khan, M. Hossain, M. Moni, Comparing different supervised machine learning algorithms for disease prediction. BMC Med. Inform. Decis. Mak. **19**, 281–281 (2019). https://doi.org/10.1186/s12911-019-1004-8
43. L. Vigoya, D. Fernandez, V. Carneiro, F. Cacheda, Annotated dataset for anomaly detection in a data center with IoT sensors. Sensors **20** (2020). https://doi.org/10.3390/s20133745