Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things

Edge-Based Blockchain Enabled Anomaly Detection for Insider Attack Prevention in Internet of Things

Yusuf Muhammad Tukur
Department of Computer Science, University of Bradford
Bradford, UK
y.m.tukur@bradford.ac.uk

Dhavalkumar Thakker
Department of Computer Science University of Bradford
Bradford, UK
d.thakker@bradford.ac.uk

Irfan-Ullah Awan
Department of Computer Science University of Bradford
Bradford, UK
i.u.awan@bradford.ac.uk

***Abstract*— Internet of Things (IoT) platforms are responsible for overall data processing in the IoT Systems. This ranges from analytics and big data processing to gathering all sensor data over time to analyze and produce long-term trends. However, this comes with prohibitively high demand for resources such as memory, computing power and bandwidth, which the highly resource constrained IoT devices lack to send data to the platforms to achieve efficient operations. This results in poor availability and risk of data loss due to single point of failure should the cloud platforms suffer attacks. The integrity of the data can also be compromised by an insider, such as a malicious system administrator, without leaving traces of their actions. To address these issues, we propose a novel Edge-based Blockchain enabled anomaly detection technique to prevent insider attacks in IoT. The technique first employs the power of edge computing to reduce the latency and bandwidth requirements by taking processing closer to the IoT nodes, hence improving availability, and avoiding single point of failure. It then leverages some aspects of sequence-based anomaly detection, while integrating distributed edge with blockchain that offers smart contracts to perform detection and correction of abnormalities in incoming sensor data. Evaluation of our technique using real IoT system datasets showed that our technique achieved the intended purpose while ensuring integrity and availability of the data which is critical to the deployment of IoT systems.**

Keywords— Internet of Things (IoT) Security, Blockchain, Insider Threat, Edge Computing, Anomaly Detection

I. INTRODUCTION

The concept of Internet of Things (IoT) is that of a paradigm by which everyday objects and devices are fortified with embedded sensors, actuators and processors providing them the ability to connect to the Internet, transfer data and communicate seamlessly with each other and with different devices. Credited to the Auto-ID Research Centre of the Massachusetts Institute of Technology (MIT), the terminology "Internet of Things" was coined in 1999 but was officially confirmed at International Telecommunications Union (ITU)'s 2005 World Summit on the Information Society (WSIS) in Tunisia [1], [2], [3], [4], [5], [6]. The core objective is to permit "autonomous and secure communication" as well as exchange of data among heterogeneous devices, services, users and applications [7], [8], [9]. Since there is no universally adopted definition for it, the term Internet of Things (IoT) has been defined differently by several different authors. It has been defined by [10]: as "an interaction between the physical and digital worlds" facilitated by "a plethora of sensors and actuators", as well as "a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object." Similarly, [11] reported it as "the development of item identifications, sensor technologies and the ability to interact with the environment." Furthermore, [12] defined it as " a highly interconnected network of heterogeneous entities such as tags, sensors, embedded devices, hand-held devices and back-end servers." Whereas according to [13], it is considered as " the latest Internet evolution that involves incorporating billions of inter-connected devices that communicate via the internet, harnessing their data and functionality to provide novel smart services and products that benefit the society." Accordingly, we have defined IoT as a huge, global, distributed network of interlinked heterogeneous devices that are "uniquely addressable based on standard communication protocols, communicating and interacting with one another in real time." [14], [1], [4], [3], [5].

There has been so much fascination for the IoT since its inception because it presents myriads of capabilities that are able to address the soaring demands of individuals, organizations and governments for improved automation, efficient processing as well as big data processing and analytics. Accordingly, a broad range of IoT applications are witnessed across many domains of our lives including manufacturing, mining, agriculture, Smart Homes, Smart Cities and Smart Healthcare among other sectors []. That is further evidenced by the way IoT is becoming increasingly conspicuous in virtually all areas of endeavor; from personal to educational, business, governmental and military applications. As of today, there are enormous uses of IoT cutting across diverse domains, as has been highlighted in different research works including [15], [14], [16], [17], [1], [18], [10], [19], [4], [15], [14], [16], [17], [1], [18], [10], [19], [4], [3], [7], [20], [9], [21], [22], [23], [6]. Table 1 summarizes and categorizes these applications under various domains. This expansive application requires the IoT systems to manage and exchange very large amounts of information including public and private safety critical data, thereby eliciting varying degrees of physical and cyber security challenges on the systems[].

In consequence, the IoT system is confronted with a multitude of threats from different kinds of security attacks across its mainly three main distinct layers: Perception, Network and Application[ref?]. The attacks usually target all or some of

the security triad of Confidentiality, Integrity and Availability (CIA) of the system[ref?]. Although the IoT shares many common features with conventional IT systems, some of its peculiarities like volume, environment and consequence make it more susceptible and thus more challenging to secure against the endlessly prevailing attacks, one of which is the insider attack. Threats from insiders can be seriously damaging and can cause immeasurable destruction and loss to governments and organizations using the IoT, because such attacks are perpetrated by persons authorized and trusted to legitimately access the system within its perimeter security. It has therefore become even more imperative to have mechanisms in place that will protect the often semi-supervised and unsupervised IoT systems from potentially harmful activities of malicious insiders[ref?].

TABLE I. IOT APPLICATION AREAS AND EXAMPLES

Domain	Example Applications
Health	medical and healthcare, smart health, health monitoring, smart healthcare, healthcare, e-health, fitness tracking and health monitoring, nursing home patient monitoring system, medical applications, medical aids, mobile healthcare, elderly assistance, smart healthcare system
Environment	climate monitoring, wildlife tracking, environment monitoring, smart environment, prediction of natural disasters, environment protection
City	smart lighting, smart parking, pollution and flood monitoring, smart cities, smart transportation, public safety, cities management, intelligent parking management, smart traffic lights, smart water systems, emergency response, emergency services, crowd monitoring, traffic management, smart security and surveillance, mobile crowd sensing, smart infrastructure, urban management, infrastructure development
Power (Energy)	smart grids, smart metering and monitoring, power grid, energy management, energy conservation, intelligence energy management, power management
Industry	industrial internet, connected vehicles, smart vehicles, smart buildings, construction management, production and assembly line management, food supply chain, production control, physical distribution, supply chain and logistics, industrial automation, building management, infrastructure monitoring, office automation, automotive
Personal & Other	Smart homes, home life, home automation, social life and entertainment, in-door navigation for the blind and visually impaired people, personal and social life, agricultural control, military, defence, modern agriculture, enhanced learning, business services

Unfortunately, there is an imbalance in the security literature, with less research has been done on the impact of malicious insiders to the IoT systems, as the great majority of the available literature focuses on other security challenges endangering the systems. This is evidenced in one of our published research works [24] as presented in Table II. The identified gaps have also motivated some of our further research in the area, including this work.

In this paper, we examine the impacts of insider attacks on a real-life IoT system prototype to observe its effects on the system, particularly the sensing data generated from the application environment. The aim is to design and develop a technique using Blockchain and edge computing technologies that trigger and inspire extensive studies that focus attention and accord due priority to finding solutions to the problems of insider threats in IoT systems which do not currently receive the attention they deserve. This work presents as its contribution an edge computing-based blockchain empowered framework to detect and correct abnormal and potentially harmful input data from sensor readings in an IoT system before the data are being transmitted to the cloud platforms for analytics and storage. The framework leverages a form of sequence-based anomaly detection technique and employs Ethereum blockchain's smart contracts to run algorithms on the edge. This set up allows to handle the detection and correction on the chain from the incoming sensor data values that are fed into the IoT system. While the distributed edge provides the hardware resources for the smart contracts to execute, it also helps in reducing latency and bandwidth as well as energy requirements for the resource constrained IoT devices to upload data directly to cloud, hence improving the processing time and data availability.

TABLE II. SECURITY THREATS TO LAYERS OF THE IOT [24]

IoT Layer	Security Threats
Perception	Node capture, malicious code injection, false data injection, replay (freshness), cryptanalysis & side channels, eavesdropping & interference, physical attack, sleep deprivation, battery draining, node jamming, exhaustion, camouflage, hardware Trojan, node replication
Network	DoS, DDoS, spoofing, sinkhole, wormhole, man in the middle, replication/replay, amplification, routing information, Sybil, side channel, masquerade, flooding, selective forwarding, data modification, repudiation of sent/received messages, malicious packet injection, de-synchronisation,
Application	Phishing, malicious worm/ virus (like Trojan), malicious scripts, selective message forwarding, data aggregation distortion, privacy breach, industrial espionage, unauthorised access, information disclosure, malware, Mirai botnet

Our framework's role is to preserve the integrity of the IoT system data thanks to the algorithms empowered by the immutability property of the blockchain which also possess certain fault-tolerance capabilities. The algorithms can screen incoming data entries using thresholds derived from the normal observations before processing and eventual storage in the cloud. The algorithms were implemented as blockchain contract in a Remix Development Environment[ref?]. We have presented and discussed our research findings in the remaining parts of this paper. Our proposed approach was evaluated using real dataset obtained from our real IoT system prototype as presented in this paper and demonstrated to achieve desired functions of adequate detection and correction as designed.

MOTVATION AND CONTRIBUTION

This work is motivated by the ever-increasing security threats to the IoT system, which ensue from its widespread applications. The main motivation however is the Insider Threats - how insiders occupy advantaged positions in the system, with which systems are further exposed and made increasingly susceptible to exploitation and attacks. The focus on anomaly detection is justified because the best approach towards effectively safeguarding any system lies in early detection and prevention of prevailing threats to, or attacks on the system[ref?].

Our main contributions in this work are:

- We established the susceptibility of typical IoT systems to insider attacks by presenting a threat model that shows how the IoT systems are vulnerable to attacks from insiders and that such attacks can be successfully achieved without being noticed by the system, and also built a live system prototype to demonstrate the same.
- We proposed and presented an edge-based blockchain enabled anomaly detection technique to address the problem, leveraging immutability of the blockchain and employing smart contracts to detect and fix abnormal entries based on sequence-based anomaly detection technique while ensuring integrity of the data
- Our technique also harnesses the integration of blockchain with edge for IoT to address the issues of latency and bandwidth, as well as the likelihood of single point of failure when dealing with the cloud platforms
- The proposed solution was evaluated using a real IoT system dataset and demonstrated to achieve accurate detection and correction on the blockchain while the integrity of the data is protected, which is vital for the overall success of the IoT.

The remainder of the paper is organized as follows: Section II discusses related work while Section III models and experiments insider threats to IoT. Edge-based anomaly detection and our proposed Blockchain empowered solutions are presented in Sections IV and V. Section VI summarizes implementation and evaluation of our proposed solution, while Section VII concludes the paper.

II. LITERATURE REVIEW

In this section, we present a comprehensive review of literatures relevant to the topic from insider threats, underlying anomaly detection techniques for our proposed framework as well as related work.

INSIDER THREATS

With increased automation and reliance on technology comes increased vulnerability and greater exposure to risk, particularly when network connectivity is involved as is the case in modern information and IoT systems.

Security risks can stem from different sources be it humans or non-humans like devices/machines; can origin internally or externaly; may occur deliberately or accidentally; and may be motivated by many factors including sabotage, data theft and destruction, fraud, hobby, spying, state sponsored crime and terrorism, political or military purposes, as well as espionage [25], [26], [27], [28].

Although there is no commonly adopted definition of insider/insider threat in the context of information security, some define it as "an insider is a human entity that has/had access to the information system of an organization and does not comply with the security policy of the organization" [29]. However, a malicious insider is a "current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems." [30], [27].

The insider threat problem has gained more prominence over time including the techniques to discover and tackle it. Among the works done to address the insider threat problem are numerous methods that have been proposed to particularly identify the conduct of potentially malicious insiders. For instance, [31] attempted to discover insider threats through the identification of abnormal behavior in enterprise social and online activity data of employees. They processed and extracted pertinent features that likely indicate the presence of insider threat behavior. However, their work focused more on insider threat activity with electronic footprints traceable to employees in the domain. It did not consider the IoT domain which is a more automated system with little human intervention and deals with real-time data; neither did the work take into account scenarios where no digital footprints are left.

The work of [32] proposed a detection method for insider threats using log analysis and event correlation. They put forward a probabilistic method to demonstrate percentage rate of event occurrence from stored log of events. The technique used the stored log file as the main input into the event correlation system to provide percentage probability of malicious insider activity. This approach is also not suitable for the IoT domain which requires a mechanism to examine incoming data before being used by the system.

Other research works include those by [33] and [34] which offered schemes defining models and frameworks to facilitate understanding of insider threats. Furthermore, researches such as [35] and [36] centered on prediction and prevention of the insider threat problem. Although some of the surveyed literature proposed ways to address the problem, none of these approaches however considered insider threat problems in the IoT domain and how they could be addressed.

On the other hand, most researches on IoT security have for a long time focused predominantly on prevention of external attacks on the system, as evidenced by [15], [14], [16], [17], [22], [37], [6], [21], [23], [12], [38], and have done little on the threats coming from insiders. As of now, only a few studies have been conducted on the topic of insider threats in IoT, and these studies have not offered solutions to the problem as it affects the IoT system. For example, the authors of [39] sought within the paper to demonstrate and categorize insider threats in relation to the IoT, showcasing attack vectors for their characterization. Their paper proposed a set of "rigorous modelling techniques" to provide a better grasp of IoT-enabled insider threat scenarios and related architectures. They applied a method of formal modelling of insider threats in the interactive theorem prover Isabelle to formalize various IoT scenarios. The output of the paper is a technique to characterize malicious and accidental insider threats to IoT systems through attack vectors, adding exactness to a provisional taxonomy through the use of a logic-based insider threat model in Isabelle.

The research carried out in [40] sought out to identify top security threats as well as evaluate existing countermeasures used to combat the threats, along with their possible applications

in IoT and multi-cloud e-Healthcare environments. They identified among many other security threats that, insider attacks could threaten the CIA security triad, and that it poses the most important risk to healthcare organizations. After conducting a systematic literature review, the authors concluded that; IoT based multi-cloud e-Healthcare organizations are vulnerable to malicious insiders among other threats that affect the CIA triad, and that malicious insider threats should be countered to ensure privacy and reliability of patients' health information using strengths of the existing security techniques and addressing the flaws.

Furthermore, [41] aimed to explore the extent to which IoT may aggravate the insider threat problem for organizations and complicate detection approaches. Therefore, they approached IoT security and privacy from an alternate viewpoint by considering the impact IoT may have on the insider threat problem. The focus of their research was on personal devices which insiders take with them and use in their offices. Although the authors stressed the importance of understanding the risks emanating from insiders in IoT environments, regrettably, only little detailed analysis of the risk has been conducted. They elaborated on attacks arising from personal IoT devices, reckoning the scope of insider attacks on consumer IoT devices and critical infrastructure. The research concluded that the security concerns in IoT are largely being assessed for external attacks only, and that current insider threat detection approaches be extended to accommodate IoT devices.

Also, the authors of [42], an extension of the work conducted in [39], considered the limits of formal modelling of infrastructures and the application of social explanation in analyzing insider threats to security and safety critical areas. They used the aviation sector as a case study following an insider attack incident which involved and led to the crash of a Germanwings flight in 2015. They studied and modelled in Isabelle the security policies and controls within passenger airplanes against insider threats, in order to show that the framework could be applied.

As may have been observed, most of the reviewed works and other similar research have either not considered insider threat in IoT systems or have neglected to proffer solutions to the problem. The few studies that have discussed insider threat in IoT mainly focused on providing techniques to facilitate understanding of the problem through formal modelling, reviewing the vulnerability of IoT to insider threats, or investigating the degree to which the IoT may worsen the insider threat problem and make detection difficult. That is, all the related literature has left a wide gap in terms of providing solutions to the insider threat in IoT taking into cognizance its distinct characteristics. The work we present in this paper tries to address this gap and ensure integrity of the IoT data, which is key to an efficient system.

ANOMALY DETECTION

Anomalies are said to be patterns in data that are not in conformity to well-defined features of normal patterns of the data. An anomaly has been defined as "*an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism*" [43]. Caused by a diversity of unusual activities ranging from insider attacks, credit card fraud and many other forms of cyber-attacks, anomalies are adjudged crucial because they point out atypical events and can trigger critical actions in extensive application areas.

Anomaly detection on the other hand is also related to discovering entries in a set of data which seem to be inconsistent with the rest of the data in that dataset [43]. It is a vital data analysis task that functions to identify such occurrences of data that do not comply with the data model; that is, it detects anomalous data from a specific dataset [44], [45]. As a better technique of the broad and dominant categories of intrusion detection, anomaly detection has to do with creating a normal behavior profile of an entity, called the "*norm profile*", against which observed behaviors of the entity are compared [46]. An entity could be a user, file, program, any parameter, host machine, etc. Anomalies are communicated by the system via an alarm, for instance, whenever observed entries/behaviors deviate from the norm profile of the entity under consideration. The original purpose of anomaly detection is to remove data entries known as outliers from a dataset, as that can adversely affect statistical analyses and automated decision making. As such, it is often seen to be made up of two stages; the *training* stage during which the *norm profile* is defined, and the *testing* stage during which the learned profile is used on the incoming data to find outliers.

Different classifications of anomaly detection techniques have been presented in previous literatures. According to [47], three major classes of anomaly detection are statistical-based, knowledge-based and machine learning-based, with each category having sub classes. While [48] also maintained these three major categories, they introduced a data mining-based class instead of the knowledge-based class. A more detailed classification can be found in the work of [49], in which the authors surveyed anomaly detection for discrete sequences and categorized it into three broad categories: *sequence-based anomaly detection,* which functions to detect atypical sequences from a database of test sequences; *contiguous subsequence-based anomaly detection,* which works to discover unusual contiguous subsequences within a long sequence; and *pattern frequency-based anomaly detection* to uncover patterns in a test sequence having abnormal rate of occurrence.

The focus of our work is not to review or discuss literature about anomaly detection, but rather to employ some of its concepts alongside other approaches in providing solutions to the identified problem in this paper. We employ the sequence-based anomaly detection[ref?] due to its applicability and relevance to the nature of IoT systems. <<In a sentence here: what does sequence-based anomaly detection mean.>>Anomaly detection for discrete sequences can be applied to different IoT application areas, including, but not limited to, critical infrastructure monitoring and protection, smart meters, smart health monitoring, smart cities, smart transportation and parking systems, smart logistics, smart retail and manufacturing as well as smart homes or smart agriculture, where sequences of sensor generated data are collected in every instance of the systems operations. Anomalies in those datasets represent cases of operational flaws or defects in the systems or its data which may be caused by accidents or malicious attacks. The attacks are often launched against automated systems like IoT to deceitfully

trigger an unintended action, such as activating a sprinkler in a laboratory or office building to damage machines or documents, shutting down a plant to minimize expected danger, diverting supply from critical equipment or launching attacks on wrong targets.

Two forms of sequence-based anomaly detection are *semi-supervised* and *unsupervised* anomaly detection. In semi-supervised anomaly detection, a normal reference dataset, or the training dataset which presumably contains only normal sequences, is initially generated. Other sequences generated from system operations, or the test sequences, are tested against the normal to detect anomalous entries. In unsupervised anomaly detection, the goal is to find abnormal sequences among an "*unlabeled database*" of sequences. Unlike the semi-supervised form, normal sequences are not generated but an anomaly value is assigned to each entry based on standard industry operations, against which the sequences are checked for anomaly.

The variants of sequence-based anomaly detection have sub-types, but that is outside the scope of this research. Therefore, we present formulations of the two techniques highlighted above.

Definition 1. The semi-supervised form is formulated below. In general application of outlier detection for semi-supervised scenario, only the training dataset for the normal sequences are supplied [50]. Hence:

Given a set of finite *(p)* sequences representing normal sensor readings from an operational IoT system, denoted by $S_n = \{s_{n1}, s_{n2}, s_{n3}, \dots, s_{np}\}$, and another set of finite *(q)* sequences for a dataset from sensor readings to be tested, denoted by $S_t = \{s_{t1}, s_{t2}, s_{t3}, \dots, s_{tq}\}$. Assign an anomaly value to the sequences in S_t with reference to the normal training sequences in S_n.

Here, the anomaly score can be a discrete or point value, a range, an expression, or the result of evaluating an expression from the normal sequences, depending on the specific nature of the problem being formulated or assumptions of the model being employed [50]. It is also important to note that *p* and *q* must not have equal lengths.

Definition 2. The unsupervised form is formulated thus:
Given a set of finite *(p)* sequences representing a dataset generated by sensor readings from a functional IoT system, which can be denoted as $S_n = \{s_{n1}, s_{n2}, s_{n3}, \dots, s_{np}\}$. Assign an anomaly score to every entry in the set S_n in relation to the rest of the sequences in S_n.

The anomaly score here may be an industry standard value from observed efficient operations[ref?].

The outliers being looked out for in typical data mining and machine learning are often frequency of (event) occurrence, standard deviation, density functions and so on among many outlier analysis/detection techniques, depending on specific requirements of the application domain. For sample applications of outlier detection in discovering measurement errors, such as in data obtained from sensors in a scientific experiment like IoT, abnormal values indicate errors, of which removal is vital in data mining and analysis tasks [50] and even more crucial in critical infrastructure domains. As such, among the common outlier detection techniques is the Z-Score or Extreme Value Analysis, which indicates the degree of deviation of a data entry from the sample's mean. Others include Linear Regression and Information Theory Models, Proximity Based Models, Probabilistic and Statistical Modeling, etc.

The problem we present in this paper is semi-supervised, so Definition 1 is for use as part of the solution we propose in this work. The anomaly score λ is a range of the maximum entries of S_n ($maxS_n$) and the minimum entries of S_n ($minS_n$), as represented in the notations below:

$$S_n = \{s_{n1}, s_{n2}, s_{n3}, \dots, s_{np}\}$$
$$maxS_n = max\{s_{n1}, s_{n2}, s_{n3}, \dots, s_{np}\}$$
$$\therefore\ max\{s_{nk} : k = 1, \dots, p\}$$

Accordingly,

$$minS_n = min\{s_{n1}, s_{n2}, s_{n3}, \dots, s_{np}\}$$
$$\therefore\ min\{s_{nk} : k = 1, \dots, p\}$$
$$\lambda := max\{s_{nk} : k = 1, \dots, p\} : min\{s_{nk} : k = 1, \dots, p\}$$
$$\therefore\ min\{s_{nk} : k = 1, \dots, p\} \geq \lambda \leq max\{s_{nk} : k = 1, \dots, p\}$$

APPROACHES USING BLOCKCHAIN AND EDGE COMPUTING

The focus of the work presented by [51] is on how decision making involving big data processing could be made optimal at edge-cloud environments. Although their research is from the viewpoint of a Software Defined Network, the authors proposed a "workload slicing scheme" facilitated by multiple edge devices, to handle what they called "data-intensive jobs" through priority-based segmentation of the input data in a cloud environment. Their proposed scheme was more to do with optimal resource utilization such as CPU, memory and storage and they argued in conclusion that it minimizes energy consumption of the total multi-edge cloud environment.

Additionally, the study conducted by [52] proposed the deployment of edge data centers in smart cities for service provisioning in vehicular environments enabled by software defined networks. They argued that their scheme would offer an optimal data flow path, improve resource allocation and utilization, as well as minimize energy consumption. Similarly, the research presented by [53] attempted to integrate cloud and edge computing with Software Defined Networks to provide a framework for edge-cloud interplay, expected to enhance secure healthcare ecosystems.

The proposed approach provided in the research published in [54] is a blockchain based edge-as-a-service framework for secure energy trading in a software defined networking-enabled vehicle-to-grid environment. Although the authors opined that processing of energy trading are taken closer to the electronic vehicles nodes thanks to edge computing and that blockchain is employed to secure the energy trading transactions across nodes, they did not make clear in the paper how their research would achieve that. Whilst the work was not clear on the kind of blockchain proposed and only explained generic blockchain

operation, the depicted blockchain process for energy trading did not clearly represent the proposed process.

The subject of the work in [55] is blockchain-based secure demand response management in smart grid systems. The scheme, as argued by the authors, can take secure energy trading decisions needed to manage overall energy loads for different sectors. They proposed a blockchain miner node selection scheme based on nodes' power consumption and processing power and presented an algorithm to achieve the selection. However, the work fell short of providing any security mechanism, not even the authentication and authorization being claimed by the authors. Rather, the authors relied on the generic operation of a blockchain with no clarity of how it can be used to achieve secure demand response management, as suggested by the paper title.

In addition to this, the authors of [56] proposed a blockchain-based energy trading scheme, "FeneChain", to handle energy trading processes in Industrial IoT with the aim of providing a secure energy trading system and enhancing energy quality. Although the work claimed secure energy trading, it mainly seems more of privacy preservation using anonymity and transparency for users. Access control was however incorporated, albeit not clarified where it occurs in the presented system model.

Furthermore, the research work presented in [57] aimed to preserve confidentiality of IoT system by proposing a blockchain-enabled distributed security framework that integrates edge-cloud and Software Defined Networking (SDN). The authors proposed a security attack detection algorithm, which they implemented at the cloud layer, claiming that it will reduce attacks at the edge, while the SDN provides dynamic, adaptive, and remote network traffic data flow services.

BLOCKCHAIN-IoT INTEGRATION

In their work presented in [58], the authors identified the need for data provenance and data integrity, which are considered the major concerns in several IoT application areas. They identified vulnerabilities in conventional IoT architecture which exposes it to attacks and proposed what they termed "The BlockPro network model" that is based on Physical Unclonable Functions (PUFs) and blockchain. Although they made it as an architecture, the model does not show either of cloud or fog, but IoT devices, smart contracts, blockchain and databases. As their contributions they presented algorithms to "enforce data provenance and data integrity" in the IoT.

Similar to [58], where to host the blockchain in IoT settings was the subject of the work carried out by [59]. They discussed hosting the blockchain as a service for IoT, comparing cloud and fog as possible candidates. They held that both cloud and fog are suitable and, after running experiments, established that fog outperforms the cloud if latency is the factor for consideration.

The use of blockchain for the IoT was surveyed by [60]. In their work, they attempted to explore ways in which blockchain features can be adapted to IoT so as to address those requirements such as seamless authentication, data privacy, robustness and ease of deployment. They discussed what they called Blockchain-based IoT (BIoT) applications, stressing the possible impact blockchain can have on conventional cloud-centered IoT applications. They also presented challenges and how possible optimizations can be made, rounding up with recommendations.

Additionally, the work of [61] proposed what they called a "novel blockchain-based distributed cloud architecture with a Software Defined Networking (SDN) to enable controller fog nodes at the edge of the network." They claimed that their model, which is a distributed cloud architecture that leverages blockchain technology, will address many IoT challenges that include availability, scalability, security, low latency, etc. they presented am architectural model called "distributed blockchain cloud architecture." However, the model replicated blockchain at both the cloud and fog, with no interoperability between the fog nodes. In the end, they presented experimental results of their work.

Furthermore, in [62], the authors proposed to design a fog computing system based on blockchain in order to avoid single point of failure due the centralization feature of the cloud. They also claimed that it can prevent IP spoofing and Sybil attack. They discussed features of both blockchain and fog computing, then made algorithms to; share transaction between fogs, control fog connected devices during downtime of one of the fog nodes, and recover the downed fog node. This shows that the fog nodes do not integrate with the blockchain. Rather, they interoperate independent of the blockchains.

The authors of [63] utilize consortium blockchain and smart contract technologies to ensure that data sharing is done only with authorization in order to achieve "secure data storing and sharing in vehicular edge networks. They also proposed a "reputation based" scheme to guarantee high-quality data sharing among vehicles. In the end, they performed analysis to show that their system provided secure data sharing and storage.

Applying a blockchain based solution to build "an open, trusted, decentralized and tamper-proof system" for Long Range Wide Area Network (LoRaWAN), thereby addressing the trust issues between it and Narrow Band IoT, is the work proposed by Using Blockchain Technology to Build [64]. They believe themselves to be the first to attempt integration of blockchain technology with LoRaWAN IoT. They discussed LoRaWAN architecture and blockchain technology, then proposed a blockchain architecture for LoRaWAN in which the blockchain system is incorporated in the network server layer of the LoRaWAN, having the servers as participants. In the end, they believed that their work provides an indisputable mechanism to verify the existence of a transactional data at some time in the network.

Adding on to this, in the paper published by [65], the authors proposed the design of a blockchain connected gateway to "adaptively and securely maintain" privacy preferences of users for IoT devices in the blockchain. They depicted a scenario where the IoT gateway connects to the blockchain network as one of its nodes, enabling it to block privacy leakages and protect users' data from unauthorized access. They also proposed a digital signature scheme to provide authentication and secure management of privacy preferences. Blockchain network was adopted as the underlying architecture to process and maintain data and resolve privacy disputes.

Contrastingly, the work of [66] attempted to use blockchain to mitigate DDoS attacks on IoT devices. They identified the constrained nature of IoT devices which makes them susceptible to cyberattacks including DDoS, and proposed the integration of IoT devices with blockchain in order to address it. They first considered the conventional architecture of the IoT, then proposed an IoT –Blockchain system model to represent their assumptions and threat model. In the end, they proposed algorithms to validate devices and instantiate IoT devices and blockchain miners. The purpose of the validation is to block rogue devices, and static resource limit is allocated to each device to help protect against DDoS attack, since the limit cannot be exceeded.

Moreover, in their work, [67] proposed a model for the security of IoT using blockchain after making a note for the security concerns in IoT and the insufficiency of existing security mechanisms to fully protect the system. They first discussed an overview of the blockchain technology and its implementation, then discussed blockchain based IoT and presented a blockchain network for IoT. Finally, they leverage on that to propose blockchain secured IoT devices.

A work in progress [68] seeks to create a proof of concept that makes low-power, resource-constrained IoT devices able to access a blockchain-based infrastructure that has the ability to meet the security and scalability needs of the ever-increasing IoT devices. To realize that, the authors configured an IoT gateway as a node of the blockchain and then proposed a messaging mechanism for low-power IoT end devices.

Further research conducted in [69] looks at the challenges of IoT with respect to security and scalability, including their weakness in resources and vulnerability to attacks. The authors also realize the capabilities of edge computing and the prospects offered by blockchain and smart contracts. To address such challenges, the authors designed and implemented and edge-IoT prototype based on blockchain and smart contracts, which they called "EdgeChain". Implementation and evaluation of the EdgeChain prototype shows that, in addition to security provisioning, the cost of integrating it with blockchain and smart contracts is reasonable. According to the authors, EdgeChain is the first of its kind to incorporate blockchain in edge computing to provide resources to various IoT applications and regulate behavior of devices without overloading them with security-related burdens.

Furthermore, in [70], the authors looked at the prospects offered by smart contracts and blockchain to decentralize cloud/fog solutions, lowering costs and enforcing predictable results without the need for any intermediary. They considered three projects that rely on blockchain for analysis and comparison, although information about the solutions adopted by the projects are scanty and hard to find. Their resultant findings described the architecture and implementation choices of the three decentralized cloud systems. In addition, they compared them and stressed the need for standardization of several features.

As researchers continue to apply blockchain in the Internet of Things, [71] analyzed their vision of a smart city and identified the demands of availability and immutability of environmental data, as well as the problems of data storage and management by the sensors. In order to address the challenges, they identified the use of blockchain as a possible candidate. Consequently, they proposed a blockchain based system, CitySense, which encourages human involvement in constructively monitoring environment quality to promote greater awareness of city health.

In another paper, [72] identified the importance of time synchronization among deployed IoT devices and the challenge to guarantee accuracy and consistency of time synchronization. To address the problem therefore, they propose a blockchain-based scheme to guarantee IoT security during time synchronization. The scheme uses the consortium blockchain representing an open but limited network environment, where the consensus process is processed by some preselected nodes that resist a number of attacks from outsiders and insiders. They presented algorithms of their scheme and produced analysis results. In the conclusion, they claimed their scheme can adapt to the changes of network topology and that time synchronization can be implemented efficiently by employing Practical Byzantine Fault Tolerance.

III. MODELING IoT INSIDER THREATS

Threats from insiders have been posing serious problems to information systems and IoT systems alike. They are one of the most difficult to prevent security threats as they have the ability to affect any system regardless of its level of security. Insider threats are mostly a deliberate, calculated and malicious breach of the system and its security, often carried out by persons with legitimate access who are usually authorized and trusted within the organization's basic and perimeter security [25], [73], [29], [30], [74], [75], [76]. The impact of this could lead to significant harm or loss to the Confidentiality, Integrity or Availability (CIA) of the affected organizational system and its data [26], [75]. Consequently, no IoT system has guaranteed security from attacks by insiders. Various threat models are in existence where, when accomplished by insiders, can have negative influence on the system. Some examples different possible threat models can be found in one of our studies [77] related to the topic. Consequently, it is important to investigate the likely impacts of those and many other threat models on the IoT systems in order to ascertain the degree of risk they pose with a view to designing appropriate countermeasures to tackle them.

In this paper, we consider a scenario of malicious insiders attempting to compromise an IoT system with full security implementation. It is a hard venture with tamper proof and tamper evident sensor nodes that authenticate to the network and transmit cryptographically encrypted data via secure communication channels to the IoT cloud infrastructure used by the applications. Nonetheless, this does not guarantee the system's protection against malicious insider attacks, as we demonstrate in this paper. As a way of compromising and bypassing the security of the system components, the malicious insiders can resort to tampering with the environment from which devices sense data and send it across to the system. The sensing devices correctly measure the tampered environment data and transmit it securely over the network to the cloud platform for processing, analytics and further use by the applications. On receiving the data, however, the IoT applications may trigger erroneous functions, give wrong

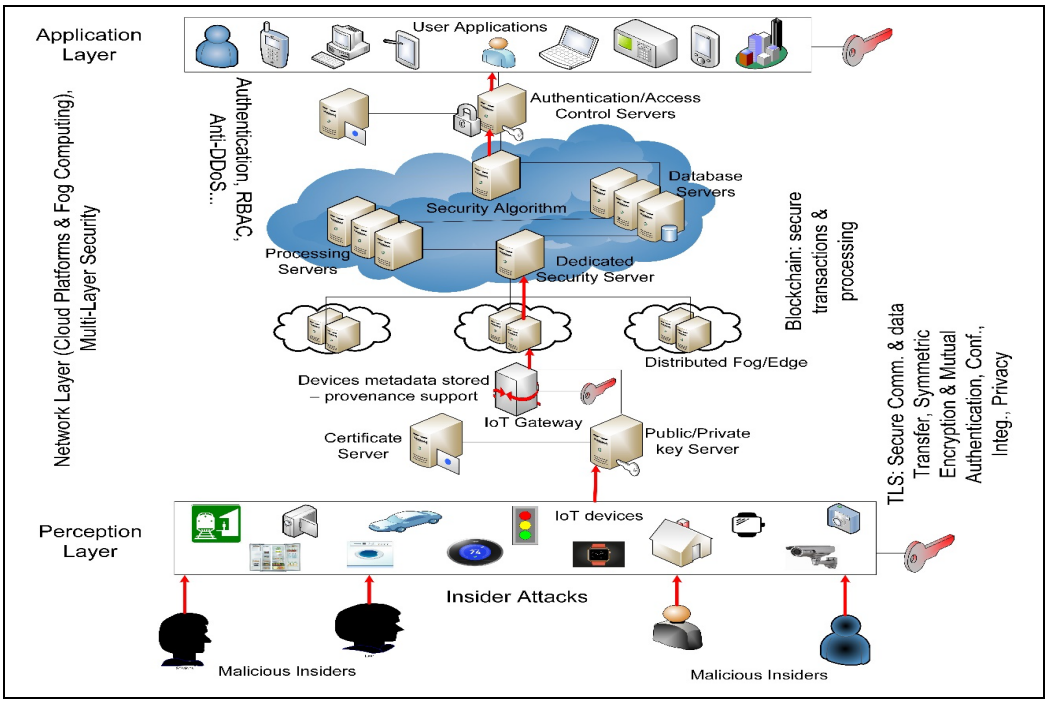

Fig. 1. Insider Attack Model on Secure IoT System

feedback commands or provide incorrect services which may be able to cause significant harm and affect the general effectiveness of the IoT system and applications.

We represent in Fig. 1 a successful insider attack scenario on a relatively secure IoT system without compromising security of the component devices. The red pointed arrows in the diagram represent the flow of incorrect data resulting from the compromised environment, where the malicious insiders made sensors to read and transmit such to the platform. With legitimate access to the system, tampering is easy for insiders as previously explained. A flowchart for the modelled scenario is also presented in Fig. 2.

Through the whole attack process, the system is tricked into assuming that it measured accurate data from the environment, and so transmitted it securely and carried out appropriate operations and functions as per predefined rule sets. That is, the security mechanisms on the system can function effectively but are unable to prevent insider attacks, because that is not incorporated into the design and implementation of the system's security architecture.

EXPERIMENTAL TOOLS AND PROCEDURE

To illustrate how the model presented in Fig. 1 works, we performed an experiment to investigate the effect of insider attacks on an IoT system.

A. Hardware Specifications

For the experiment, we obtained a Laptop PC running Windows 10 Home 64-bit Operating System with Installed memory (RAM) of 8.00 GB, 500 GB Hard Disk Drive and Processor details: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz. We also obtained Marvin, an IoT developer board built to work and securely connect with LoRaWAN on any Universal Serial Bus (USB) port. The Marvin is distributed with Arduino Leonardo bootloader and is programmed using Arduino software to read and send sensor data over LoRaWAN, routing it through the required platform. Then, we set up a functioning LoRa-based IoT System prototype with sensors, by programming and configuring the device in Arduino to join and send data over a LoRaWAN gateway on The Things Network.

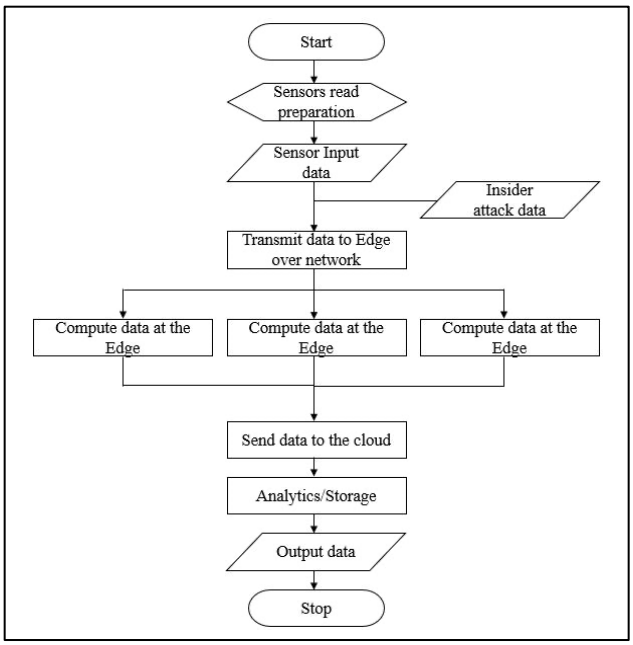

Fig. 2. Insider Attack Modelling Flowchart

The device joined the network and was successfully activated using over the air activation (OTAA) to send collected sensor data to the network. That was achieved by connecting the device to a laptop Personal Computer (PC) for device configuration, programming, and data logging.

B. Software Specifications

The main software used in the experiment is Arduino Integrated Development Environment (IDE), version 1.8.8 Windows Installer for Windows. However, since the Arduino serial monitor is not able to store or log sensor readings, we therefore used a third party application known as Cool Term for that purpose; since it has the ability to save both absolute or relative sensor readings along with timestamps and in different file formats.

C. Experimental Set Up

At first, we installed the latest version of Arduino IDE for our Operating System to program the device. We then configured the device to join and send data over a LoRaWAN gateway on The Things Network. To achieve that, we created an application in the Things Network console and registered our device. These processes provided us all the needed security credentials and activation parameters to establish secure connection to the network. The generated credentials were used to configure our Marvin device in the Arduino IDE to interact with the network and send data. We then connected our developer board to the PC with a Grove Temperature Sensor as shown in Fig. 3 and completed the configuration. Our study published in [78] provides more detailed information about IoT system settings and configuration.

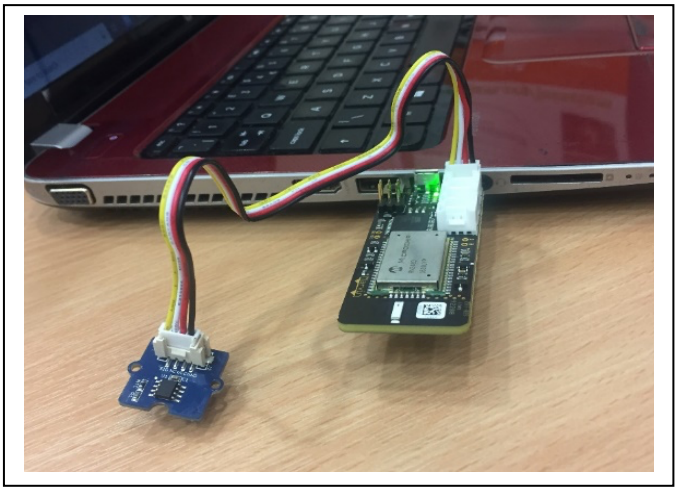

Fig. 3. Setting IoT Device Parameters to join LoRaWAN

EXPERIMENTS AND RESULTS

For the insider attack experimentation, we first checked the state of the IoT prototype we have set up. The system worked very well, reading and securely sending data to the platform for analytics. It is a relatively secure system because the component elements involved implement their own levels of security which make it difficult to breach. Nevertheless, beyond the security of components, persons with legitimate access to the system pose huge threats and the system is at risk if such persons have malicious intent to compromise it for financial gain or any other motive. Therefore, we set up six different experiments in two locations; three each at indoors and outdoors, and recorded the actual data in each case, as highlighted below. For both locations, we considered and executed a threat model where an insider tampers not with the sensing devices or the network because they are tamper proof and encrypted respectively, but with the physical surrounding about which the sensors read and send data. As such, the sensor and the entire system work effectively, unhindered. Despite this, the integrity of the data captured and sent over the network to the platform has been compromised by malicious insiders who succeeded in altering the properties around the perception devices.

For the specific attacks, we isolated the sensing device to create a favorable attack environment to achieve a compromise. We then injected low temperature into it for a considerable amount of time, captured and logged the data. We did the same by injecting high temperature into the isolated environment for about the same time and recorded the data. Fig. 4 and Fig. 5 respectively show the results of performing the described processes in both indoor and outdoor settings. It can be seen from Fig. 4 that while the normal temperature is almost uniform, fluctuating between 18 and 23 degree Celsius, the recorded successful cold and heat attacks by insiders on the system fluctuate rather abnormally with obvious effects. The injected heat can be seen to make the system record as high as 52 degree Celsius while the cold temperature attack by malicious insiders has made the system record as low as -4 degree Celsius. Similarly, results of Fig. 5 show an actual outdoor temperature value of between 15 and 21 degree Celsius while insider cold and heat attacks recorded peak values of -6 and 51 respectively.

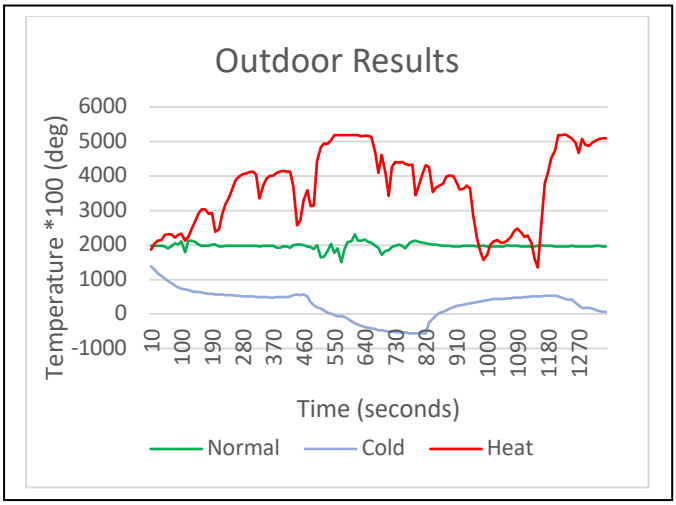

Fig. 4. Experiment Results - Outdoor

Measuring the temperature in itself is not the main aim of this experiment, the purpose is rather to investigate and show the vulnerability of IoT systems to insider attacks and how the attacks can be performed successfully on an IoT system for which temperature is a vital property. The experiment is therefore relevant and useful for many application domains of IoT such as industrial temperature monitoring, critical

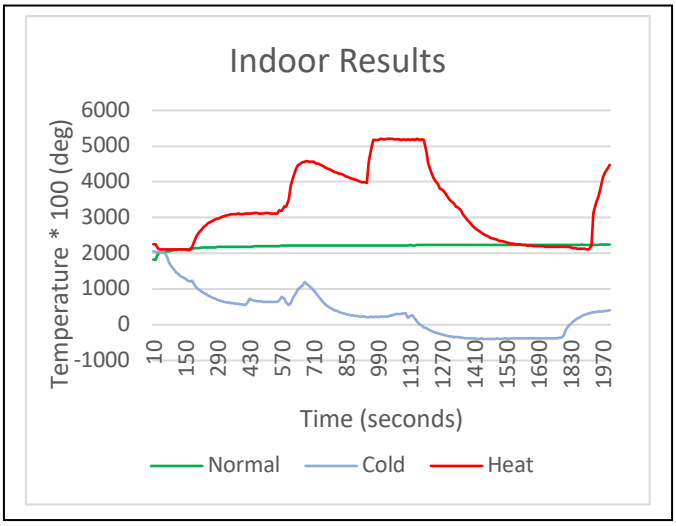

Fig. 5. Insider Attack Experiment Results - Indoor

infrastructure monitoring such as smart grid or power systems, fire-fighting systems, and so on. The experiment is therefore relevant and useful for many application domains of IoT such as industrial temperature monitoring, critical infrastructure monitoring such as smart grid or power systems, fire-fighting systems, and so on. These and many other domains exist where the demonstrated attacks could cause significant damage. It may have been observed from the results that the system responded adequately to the anomalies due to attacks on its environment, read and sent compromised data over the network for analytics and processing without taking any clue as to the authenticity or integrity of such data.

Consequently, it has become imperative to trigger more research and focus attention on insider attacks and ensuing anomaly with data in the area so that appropriate solutions can be provided to ensure more secure and efficient IoT systems. Hence, we have presented our solution in the section that follows.

EDGE-BASED ANOMALY DETECTION

We have established through research and experimentation in the previous sections that insider threats pose huge risks to IoT systems. The threats are also of great concerns because they penetrate through normal system and component devices' security. To address the problem, we therefore designed and implemented an anomaly detection framework that runs at the edge level to detect and prevent the system from the negative effects of insider threats by ensuring integrity and correctness of sensor generated data that go into the system. Our framework, as depicted in Fig. 6, is based on the three-layered IoT architecture we have used to model insider attacks on IoT systems. The main function of the framework takes place at the edge and it can be seen from the figure that all the edge nodes are part of the anomaly detection and correction stage. Within the framework, the anomaly detection stage is at the edge, between the sensing layer and the cloud platform. In essence, its focus is to function such that, whenever data is being sensed from the environment and transmitted into the IoT system, it goes through the edge nodes first. Each of these edge nodes run anomaly detection and correction algorithms we have developed to assess the incoming data against known normal values for any abnormality, log any found abnormality into the Anomalous Data Log, then correct the abnormality using generated normal data benchmarks before processing at the edge or transmission to the cloud for big data processing, analytics and storage.

The functionality deployed and executed on the edge nodes is facilitated by a series of three algorithms that collectively

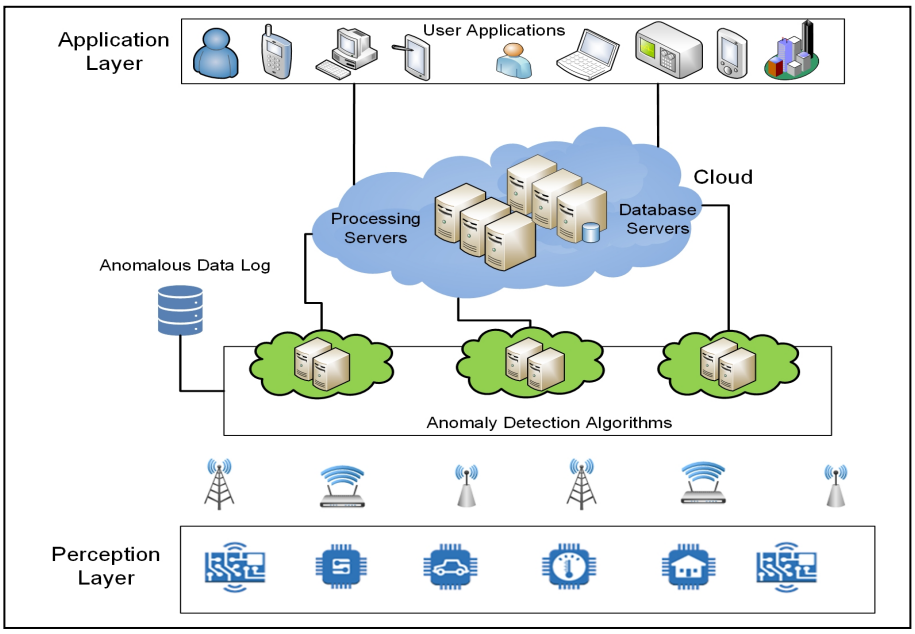

Fig. 6. Edge-Based Anomaly Detection Framework for Internet of Things

produced an effective anomaly detection and correction mechanism that also keeps records of all anomalous data for later analysis and provenance. At first, Algorithm 1 analyses the intervals in incoming sensor data values and screens it against the normal, to observe any deviations from the standard values being read by sensors under normal operating conditions. That is an important step towards detecting abnormality in the system data so that it can be promptly addressed. On the other hand, Algorithm 2 extends beyond just the intervals of sensor data values to introduce minimum and maximum functions for the array of normal values. It operates such that if the intervals between two successive elements of the array of incoming sensor data values is greater than the maximum interval of any two successive elements in the whole array of normal data values, then the value greater than normal maximum is pushed into the Anomalous Data Log and reassigned to the normal maximum value already obtained. Similarly, if the intervals between two successive elements of the array of incoming sensor data values is less than the minimum interval of any two successive elements in the entire array of normal data values, then the value less than the normal minimum is pushed into the Anomalous Data Log and reassigned to the normal minimum value already obtained. Similarly, if the intervals between two successive elements of the array of incoming sensor data values is less than the minimum interval of any two successive elements in the entire array of normal data values, then the value less than the normal minimum is pushed into the Anomalous Data Log and reassigned to the normal minimum value already obtained.

Algorithm 1: Analyzing Intervals in Sensor Data

```
Sensor reads and inputs data Si
Transmits Si to the fog over network
Fog receives Si, i = 1, 2, 3 … n
Store list of data values in array data []
declare data [] = {S1, S2, S3 … Sn}
analyse the difference between data []
  elements and store in diff [] array
for (int i = 1; i < data.length; i++){
       declare Di = Si - Si-1
       push D to diff []
}
```

Algorithm 1: Analysing Sensor Data for Anomaly

Algorithm 3 is a hybrid that combines Algorithms 1 and 2 and refines it further by introducing benchmarks, which are values obtained based on standard operating procedures of specific application domains. The essence of the benchmarking is to ensure that no suspicious data value above the upper benchmark or below the lower benchmark of the standard normal data values goes through the system. That is, any data value outside of the industry standard normal range is flagged as potentially harmful and sent to the Anomalous Data Log, then reassigned to the upper or lower benchmark value based on the screening outcome. For implementation and evaluation purposes, the upper and lower benchmark values can be set

Algorithm 2: Fixing Anomalies using Training Data

```
Sensor reads and inputs data Si
Transmits Si to the fog over network
Fog receives Si, i = 1, 2, 3 … n
Store list of data values in array data []
declare data [] = {S1, S2, S3 … Sn}
Set maximum = max [S1:Sn] //normal max
Set minimum = min [S1:Sn] //normal min
analyse the difference between elements
  and store in diff [] array
for (int i = 1; i < data.length; i++){
       declare Di = Si - Si-1
       if Di > maximum then
              Log Di in database
              Di = maximum
       else if Di < minimum then
              Log Di in database
              Di = minimum
       else          Di
       push Di to diff []
}
```

Algorithm 2: Fixing Anomalies Based on Normal Training Data

according to known industry standards respectively or to the results of evaluating an expression formulated for the problem. Therefore, any data values outside of the range will be blocked and logged, then reassigned to either the industry standards or evaluated expressions. Hence, all incoming sensor data values are first screened and checked against the standard benchmarks. If a value goes higher than the upper benchmark, then the higher value is sent to the Anomalous Data Log and the upper benchmark value is assigned to replace it. Else if a value falls below the lower benchmark, then the lower value is pushed and stored in the Anomalous Data Log and the lower benchmark value is reassigned to it. Thereafter, the algorithm continues executing to analyze the intervals of incoming sensor data for anomalies and then correct any of such anomalies as presented by Algorithms 1 and 2.

The control measures provided in the algorithms have been implemented and tested and are found to work effectively and produce desired outcome. These raise growing concerns for secure processing of the collected and transmitted device data that are often sensitive. Consequently, fog computing is introduced in our proposed solution to address the shortcomings. The fog is important in this solution because it ensures availability by taking processing closer to the edge of the nodes; thereby reducing latency and bandwidth requirements of the resource constrained IoT devices. It also provides distribution, which prevents single point of failure should anything go wrong in the cloud. However, the fog cannot guarantee data integrity and is equally exposed to the vulnerabilities of the cloud.

IV. PROPOSED BLOCKCHAIN SOLUTION

We have noted the vulnerability of cloud services to the possibility of being compromised by, for instance, the system administrator, which poses great risk to the sensitive IoT data

they handle and even greater risk of harm arising from their susceptibility to single point of failure or DDoS attacks, that can affect service availability. These raise growing concerns for secure processing of the collected and transmitted device data that are often sensitive. As a result, fog computing is introduced in our proposed solution to address the shortcomings. The fog is important in this solution because it ensures availability by taking processing closer to the edge of the nodes; thereby reducing latency and bandwidth requirements of the resource constrained IoT devices. It also provides distribution which prevents single point of failure should anything go wrong in the cloud. However, the Fog cannot guarantee data integrity and is equally exposed to the vulnerabilities of the cloud. Consequently, it becomes important to leverage the Ethereum blockchain technology and integrate it with the Fog in order to ensure transparency and immutability, a feature which prevents tampering with the data even by a malicious database administrator who could tamper with data stored on the cloud.

Algorithm 3: Anomaly Detection & Correction

```
Sensor reads and inputs data Si
Transmits Si to the fog over network
Fog receives Si, i = 1, 2, 3 … n
Set upper benchmark
Set lower benchmark
Store list of data values in array data []
for (int i = 0; i ≤ n; i++){
       if Si > upper benchmark then
              Log Si in database
              Si = upper benchmark
       else if Si < lower benchmark then
              Log Si in database
              Si = lower benchmark
       else
              Si
       push Si to data []
}
declare data [] = {S1, S2, S3 … Sn}
Set maximum = max [S1:Sn] //normal max
Set minimum = min [S1:Sn] //normal min
analyse the difference between elements
  and store in diff [] array
for (int i = 1; i < data.length; i++){
       declare Di = Si - Si-1
       if Di > maximum then
              Log Di in database
              Di = maximum
       else if Di < minimum then
       `      Log Di in database
              Di = minimum
       else
              Di
       push Di to diff []
}
```

Algorithm 3: Anomaly Detection and Correction

Our framework, presented in Fig. 7, represents the three-layered architecture of IoT comprising the perception layer, the network layer and the application layer. The solution in this work is deployed between the perception and the network layer. As it can be seen from the framework, all the Edge/Fog nodes have Ethereum deployed on them which makes each to be a node of the larger blockchain network. A full Ethereum node deploys well on a laptop computer; hence it is feasible to deploy it on the Edge/Fog instance? as shown in the framework. That means apart from performing the required computations timely, the edge computing nodes have additional capabilities powered by the Ethereum Smart Contracts to carry out integrity checks on all incoming data before allowing it through to the ledger for storage and onward use by the applications. That is because our framework recognized the fact that integrity of collected data is vital for the success of every IoT system. Therefore, data integrity must be preserved in order to ensure correctness and efficiency of the system. The smart contracts allow for set of rules, programs and algorithms to be written and implemented [70], [60], [69], [79] to provide the required security safeguards against actions that could affect the confidentiality, integrity or availability of the system. Therefore, for the role blockchain will play in the solution, we will instead write the algorithms in a smart contract which will carry out integrity tests on incoming data and deny faulty values entry through the system by correcting the input data based on standard operating conditions and log the faulty inputs with timestamp for review and provenance.

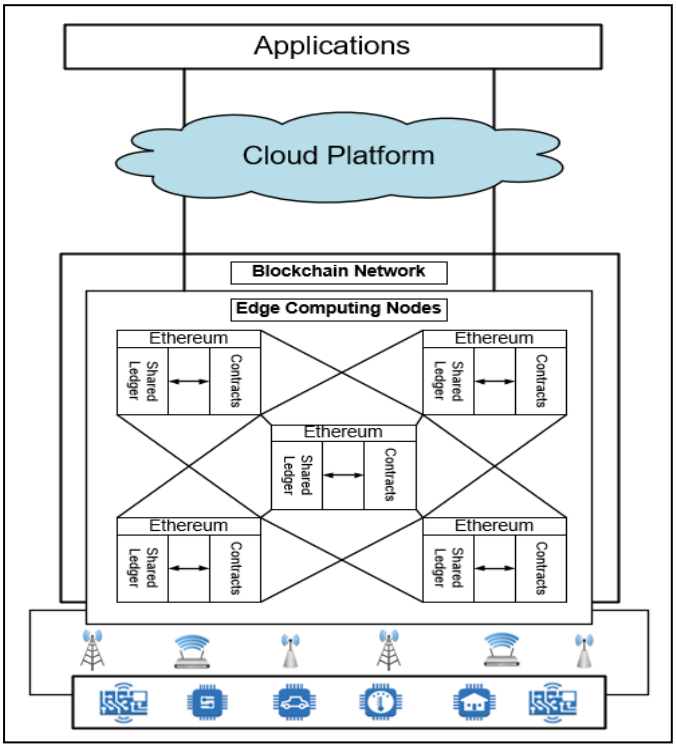

Fig. 7. IoT-Blockchain Framework

V. IMPLEMEMTATION AND EVALUATION

Our proposed solution has been implemented in Ethereum blockchain, which offers Ethereum Virtual Machine (EVM) that serves as runtime environment for our smart contracts, on Rinkeby Test Network, using the MetaMask browser plugin that

allows creation of Wallets and helps to connect to either a test network or the main network. The Smart Contract for the solution was written in Remix, a browser based Integrated Development Environment for writing and implementing contracts in the Solidity language. The Smart Contract was developed in line with the algorithms presented in this paper, which are based on the anomaly detection and correction technique formulated in Section II. The contract was compiled and successfully deployed within the Remix environment. For implementation purposes, we deployed the contract on the JavaScript Virtual Machine environment through a test blockchain account, which provides up to 100 ether allowance and some default gas limit from which to fuel execution of the transactions within the contract. On deployment, the transactions in the contract get mined and executed successfully, from which credentials such as transaction hash, contract address, gas limit, as well as transaction and execution costs are derived. The statistics provide information about the contract and its status of deployment and can thus help in evaluating the effectiveness of the contract. Fig. 8, Fig. 9, and Fig. 10 present some steps of the contract creation and deployment.

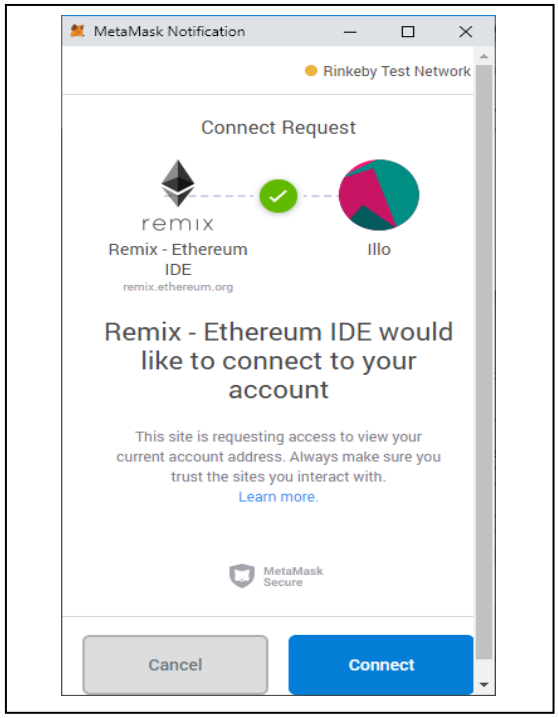

Fig. 8. Linking Blockchain Account to Ethereum IDE

To evaluate the proposed solution, we used two real-world datasets generated from our IoT experiments, which have been explained in Section III. The first dataset is for the indoor running of the experiment where the normal sensor readings, as per our earlier formulation of the anomaly detection technique, is used as the training dataset against which subsequent sensor data are tested. The second dataset is for similar outdoor running of the experiment. In both cases, two attacks each were performed on the system and the attack datasets were generated, which were passed to our proposed system for detection and correction purpose. Therefore, our system was evaluated for

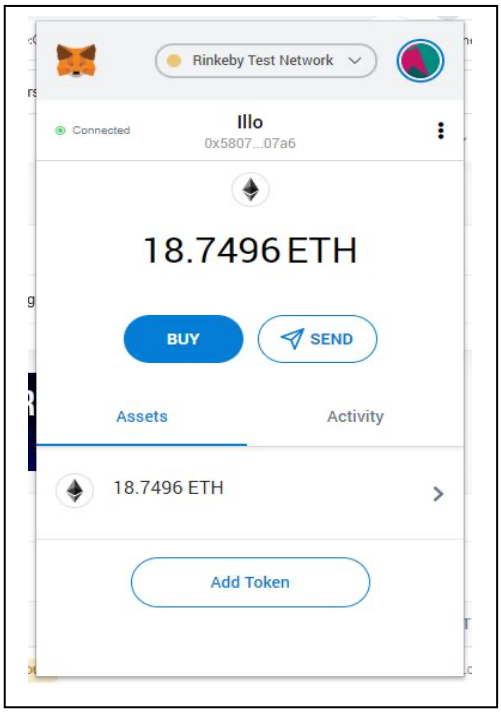

Fig. 9. Account with Wallet balance linked to Test Network

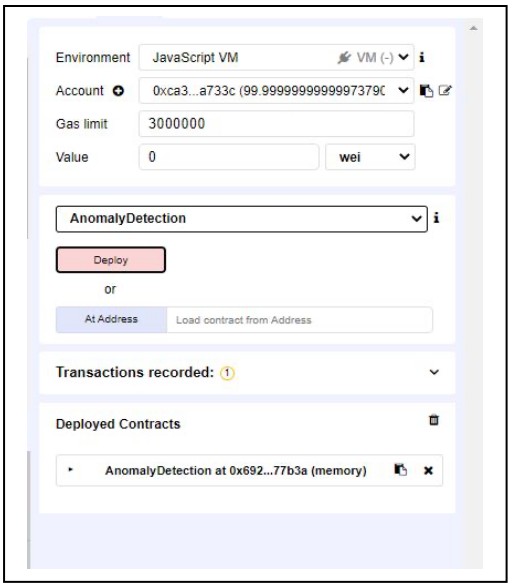

Fig.10. Deployed Anomaly Detection Contract

anomalies in four different attack datasets based on real sensors data. For the experimental evaluation presented in this paper, we arbitrarily took 100 entries from each of the four attack datasets and tested each entry against the entire training dataset of the corresponding experiment location, indoor or outdoor, so as to ensure more in-depth detection. As described in our formulation earlier, the training and test dataset sequences must not be equal in length.

After running the evaluation, the results as presented in the charts of Fig. 11 below show that our proposed solution works

effectively as designed, detecting and correcting divergent sensor data to protect the system.

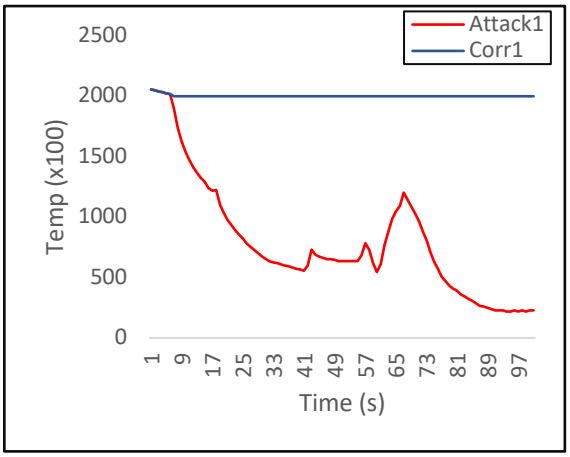

Fig. 11(a)

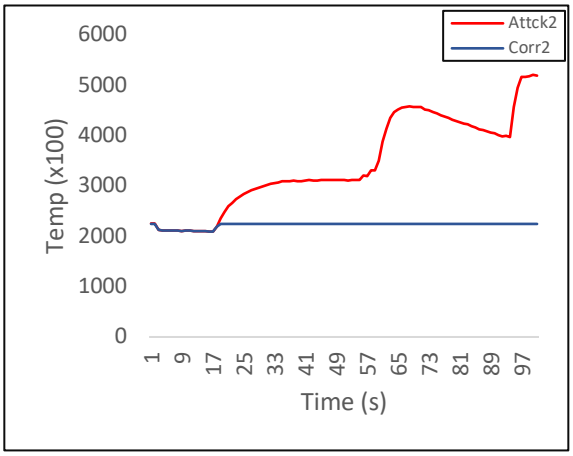

Fig. 11(b)

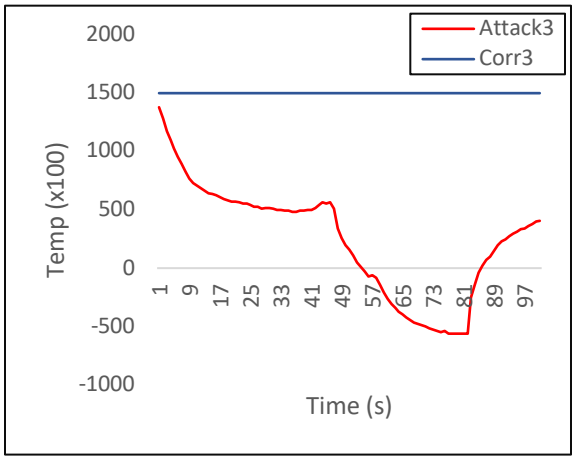

Fig. 11(c)

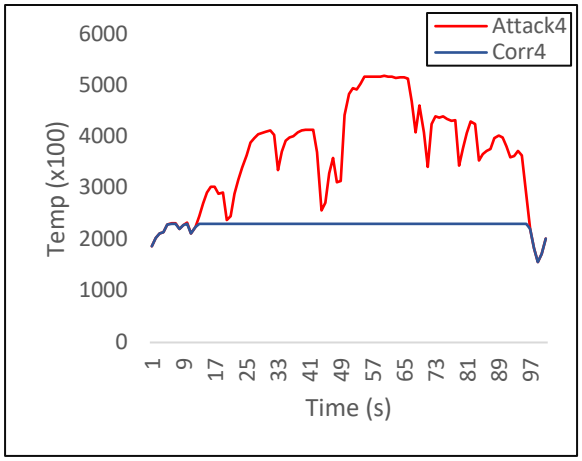

Fig. 11(d)

Fig. 11. Anomaly Detection and Correction Accuracy of our Proposed Blockchain Enabled Smart Contract Solution. The charts (a) – (d) show detection and correction efficiency of our solution, for four different attack scenarios of the experiments presented earlier in this work.

From the viewpoint of the overall system, the anomaly detection accuracy can be measured using true positives, which are the number of correctly identified abnormal entries and true negatives, which are the number of correctly identified normal entries indicators as derived during evaluation [80]. Furthermore, evaluation of our approach shows how the integrity of the data entries are preserved after executing the contract that detects and corrects anomalous values. That is achieved by assigning unique hash to every transaction which is also linked to the blockchain account of the contract to enforce immutability. Our contract deployment also showed very little gas consumption from the default gas limit provided by Remix for contract deployments.

VI. CONCLUSION AND FUTURE WORK

The IoT is continuously becoming an integral part of our lives, handling very large-scale sensitive, private and safety critical information, which makes securing it a matter of high precedence. Edge computing offers an efficient mechanism to collect data and from IoT environment and perform timely processing closer to the nodes. Blockchain on the other hand provides secure transaction processing and ensures the integrity of the data is protected. Merging the two together provide improved capabilities to help the IoT system achieve secure, accurate and timely data collection and processing, which is very imperative for the success of the IoT especially in time sensitive and critical infrastructure domains. Through this work, we have presented an Edge-based Blockchain empowered solution to detect and correct anomalous data values within IoT environment and ensure its effective operations. We evaluated the proposed solution using real data from a live IoT experiment and the results show that the solution works accurately, ensuring secure and timely transaction processing and maintaining data integrity for the IoT system. Our future work intends to investigate possible deployment of machine learning and data mining techniques on Blockchain to perform full-fledged anomaly detection, with its prospects and challenges.

References

1. Handong, Z. and Z. Lin. *Internet of Things: Key technology, architecture and challenging problems*. in *2011 IEEE International Conference on Computer Science and Automation Engineering*. 2011.
2. Ning, H. and Z. Wang, *Future Internet of Things Architecture: Like Mankind Neural System or Social Organization Framework?* IEEE Communications Letters, 2011. **15**(4): p. 461-463.
3. Atzori, L., A. Iera, and G. Morabito, *The Internet of Things: A survey.* Computer Networks, 2010. **54**(15): p. 2787-2805.
4. Gubbi, J., et al., *Internet of Things (IoT): A vision, architectural elements, and future directions.* Future Generation Computer Systems, 2013. **29**(7): p. 1645-1660.
5. Yakubu, O., O. Adjei, and B.C. Narendra, *A Review of Prospects and Challenges of Internet of Things.* International Journal of Computer Applications, 2016. **139**(10).
6. Misra, S., et al. *A Learning Automata Based Solution for Preventing Distributed Denial of Service in Internet of Things*. in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*. 2011.
7. Khan, R., et al. *Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*. in *2012 10th International Conference on Frontiers of Information Technology*. 2012.
8. Whitmore, A., A. Agarwal, and L. Da Xu, *The Internet of Things—A survey of topics and trends.* Information Systems Frontiers, 2015. **17**(2): p. 261-274.
9. Zanella, A., et al., *Internet of Things for Smart Cities.* IEEE Internet of Things Journal, 2014. **1**(1): p. 22-32.
10. Sethi, P. and S.R. Sarangi, *Internet of Things: Architectures, Protocols, and Applications.* Journal of Electrical and Computer Engineering, 2017. **2017**.
11. Weber, R.H., *Internet of things: Privacy issues revisited.* Computer Law & Security Review, 2015. **31**(5): p. 618-627.
12. Malina, L., et al., *On perspective of security and privacy-preserving solutions in the internet of things.* Computer Networks, 2016. **102**(Supplement C): p. 83-95.
13. Jayaraman, P.P., et al., *Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation.* Future Generation Computer Systems, 2017. **76**(Supplement C): p. 540-549.
14. Kozlov, D., J. Veijalainen, and Y. Ali. *Security and privacy threats in IoT architectures*. in *Proceedings of the 7th International Conference on Body Area Networks*. 2012. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
15. Yang, Y., et al., *A Survey on Security and Privacy Issues in Internet-of-Things.* IEEE Internet of Things Journal, 2017. **PP**(99): p. 1-1.
16. Nia, A.M. and N.K. Jha, *A Comprehensive Study of Security of Internet-of-Things.* IEEE Transactions on Emerging Topics in Computing, 2017. **PP**(99): p. 1-1.
17. Lin, J., et al., *A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications.* IEEE Internet of Things Journal, 2017. **PP**(99): p. 1-1.
18. Datta, S.K., C. Bonnet, and N. Nikaein. *An IoT gateway centric architecture to provide novel M2M services*. in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. 2014.
19. Al-Fuqaha, A., et al., *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications.* IEEE Communications Surveys & Tutorials, 2015. **17**(4): p. 2347-2376.
20. Sun, Y., et al., *Internet of Things and Big Data Analytics for Smart and Connected Communities.* IEEE Access, 2016. **4**: p. 766-773.
21. Atamli, A.W. and A. Martin. *Threat-Based Security Analysis for the Internet of Things*. in *2014 International Workshop on Secure Internet of Things*. 2014.
22. Nawir, M., et al. *Internet of Things (IoT): Taxonomy of security attacks*. in *2016 3rd International Conference on Electronic Design (ICED)*. 2016.
23. Alaba, F.A., et al., *Internet of Things security: A survey.* Journal of Network and Computer Applications, 2017. **88**: p. 10-28.
24. Tukur, Y.M., D. Thakker, and I. Awan. *Multi-layer Approach to Internet of Things (IoT) Security*. in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*. 2019.
25. Warkentin, M. and R. Willison, *Behavioral and policy issues in information systems security: the insider threat.* European Journal of Information Systems, 2009. **18**(2): p. 101-105.
26. Mylrea, M., et al. *Insider Threat Cybersecurity Framework Webtool & Methodology: Defending Against Complex Cyber-Physical Threats*. in *2018 IEEE Security and Privacy Workshops (SPW)*. 2018.
27. Claycomb, W.R., et al. *Identifying indicators of insider threats: Insider IT sabotage*. in *2013 47th International Carnahan Conference on Security Technology (ICCST)*. 2013.
28. Sarkar, K.R., *Assessing insider threats to information security using technical, behavioural and organisational measures.* information security technical report, 2010. **15**(3): p. 112-133.
29. Kandias, M., et al. *An insider threat prediction model*. in *International Conference on Trust, Privacy and Security in Digital Business*. 2010. Springer.
30. Mundie, D.A., S. Perl, and C.L. Huth. *Toward an Ontology for Insider Threat Research: Varieties of*

Insider Threat Definitions. in *2013 Third Workshop on Socio-Technical Aspects in Security and Trust*. 2013.
31. Gavai, G., et al. *Detecting insider threat from enterprise social and online activity data*. in *Proceedings of the 7th ACM CCS international workshop on managing insider security threats*. 2015. ACM.
32. Ambre, A. and N. Shekokar, *Insider threat detection using log analysis and event correlation.* Procedia Computer Science, 2015. **45**: p. 436-445.
33. Hunker, J. and C.W. Probst, *Insiders and Insider Threats-An Overview of Definitions and Mitigation Techniques.* JoWUA, 2011. **2**(1): p. 4-27.
34. Nurse, J.R., et al. *Understanding insider threat: A framework for characterising attacks*. in *2014 IEEE Security and Privacy Workshops*. 2014. IEEE.
35. Hugl, U. *Putting a hat on a Hen? Learnings for malicious insider threat prevention from the background of German white-collar crime research*. in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. 2015. Springer.
36. Hoyer, S., et al. *Fraud prediction and the human factor: An approach to include human behavior in an automated fraud audit*. in *2012 45th Hawaii International Conference on System Sciences*. 2012. IEEE.
37. Zhang, C. and R. Green. *Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network*. in *Proceedings of the 18th Symposium on Communications & Networking*. 2015. Society for Computer Simulation International.
38. Saied, Y.B., et al., *Lightweight collaborative key establishment scheme for the Internet of Things.* Computer Networks, 2014. **64**(Supplement C): p. 273-295.
39. Kammüller, F., J.R. Nurse, and C.W. Probst. *Attack tree analysis for insider threats on the IoT using Isabelle*. in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. 2016. Springer.
40. Ahmed, A., et al., *Malicious insiders attack in IoT based multi-cloud e-healthcare environment: a systematic literature review.* Multimedia Tools and Applications, 2018. **77**(17): p. 21947-21965.
41. Nurse, J.R., et al. *Smart insiders: exploring the threat from insiders using the internet-of-things*. in *2015 International Workshop on Secure Internet of Things (SIoT)*. 2015. IEEE.
42. Kammüller, F. and M. Kerber. *Investigating airplane safety and security against insider threats using logical modeling*. in *2016 IEEE Security and Privacy Workshops (SPW)*. 2016. IEEE.
43. Henrion, M., et al., *CASOS: a subspace method for anomaly detection in high dimensional astronomical databases.* Statistical Analysis and Data Mining: The ASA Data Science Journal, 2013. **6**(1): p. 53-72.
44. Garg, S. and S. Batra, *A novel ensembled technique for anomaly detection.* International Journal of Communication Systems, 2017. **30**(11): p. e3248.
45. Ahmed, M., A.N. Mahmood, and J. Hu, *A survey of network anomaly detection techniques.* Journal of Network and Computer Applications, 2016. **60**: p. 19-31.
46. Ye, N. and Q. Chen, *An anomaly detection technique based on a chi - square statistic for detecting intrusions into information systems.* Quality and Reliability Engineering International, 2001. **17**(2): p. 105-112.
47. Garcia-Teodoro, P., et al., *Anomaly-based network intrusion detection: Techniques, systems and challenges.* computers & security, 2009. **28**(1-2): p. 18-28.
48. Patcha, A. and J.-M. Park, *An overview of anomaly detection techniques: Existing solutions and latest technological trends.* Computer networks, 2007. **51**(12): p. 3448-3470.
49. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection for discrete sequences: A survey.* IEEE transactions on knowledge and data engineering, 2010. **24**(5): p. 823-839.
50. Kriegel, H.-P., P. Kröger, and A. Zimek, *Outlier detection techniques.* Tutorial at KDD, 2010. **10**: p. 1-76.
51. Aujla, G.S., et al., *Optimal Decision Making for Big Data Processing at Edge-Cloud Environment: An SDN Perspective.* IEEE Transactions on Industrial Informatics, 2018. **14**(2): p. 778-789.
52. Aujla, G.S.S., et al., *EDCSuS: Sustainable Edge Data Centers as a Service in SDN-enabled Vehicular Environment.* IEEE Transactions on Sustainable Computing, 2019: p. 1-1.
53. Aujla, G.S., et al., *SAFE: SDN-Assisted Framework for Edge–Cloud Interplay in Secure Healthcare Ecosystem.* IEEE Transactions on Industrial Informatics, 2019. **15**(1): p. 469-480.
54. Jindal, A., G.S. Aujla, and N. Kumar, *SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment.* Computer Networks, 2019. **153**: p. 36-48.
55. Jindal, A., et al., *GUARDIAN: Blockchain-based Secure Demand Response Management in Smart Grid System.* IEEE Transactions on Services Computing, 2019: p. 1-1.
56. Li, M., et al., *Blockchain-Enabled Secure Energy Trading With Verifiable Fairness in Industrial Internet of Things.* IEEE Transactions on Industrial Informatics, 2020. **16**(10): p. 6564-6574.
57. Medhane, D.V., et al., *Blockchain-Enabled Distributed Security Framework for Next-Generation IoT: An Edge Cloud and Software-Defined Network-Integrated Approach.* IEEE Internet of Things Journal, 2020. **7**(7): p. 6143-6149.

58. Javaid, U., M.N. Aman, and B. Sikdar, *BlockPro: Blockchain based Data Provenance and Integrity for Secure IoT Environments*, in *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*. 2018, ACM: Shenzhen, China. p. 13-18.
59. Samaniego, M. and R. Deters. *Blockchain as a Service for IoT*. in *2016 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)*. 2016.
60. Fernández-Caramés, T.M. and P. Fraga-Lamas, *A Review on the Use of Blockchain for the Internet of Things.* IEEE Access, 2018. **6**: p. 32979-33001.
61. Sharma, P.K., M. Chen, and J.H. Park, *A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT.* IEEE Access, 2018. **6**: p. 115-124.
62. Jeong, J.W., B.Y. Kim, and J.W. Jang, *Security and Device Control Method for Fog Computer using Blockchain*, in *Proceedings of the 2018 International Conference on Information Science and System*. 2018, ACM: Jeju, Republic of Korea. p. 234-238.
63. Kang, J., et al., *Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks.* IEEE Internet of Things Journal, 2018: p. 1-1.
64. Lin, J., Z. Shen, and C. Miao, *Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT*, in *Proceedings of the 2nd International Conference on Crowd Science and Engineering*. 2017, ACM: Beijing, China. p. 38-43.
65. Cha, S., et al., *A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things.* IEEE Access, 2018. **6**: p. 24639-24649.
66. Javaid, U., et al., *Mitigating loT Device based DDoS Attacks using Blockchain*, in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*. 2018, ACM: Munich, Germany. p. 71-76.
67. Singh, M., A. Singh, and S. Kim. *Blockchain: A game changer for securing IoT data*. in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. 2018.
68. Özyılmaz, K.R. and A. Yurdakul. *Work-in-progress: integrating low-power IoT devices to a blockchain-based infrastructure*. in *2017 International Conference on Embedded Software (EMSOFT)*. 2017.
69. Pan, J., et al., *EdgeChain: An Edge-IoT Framework and Prototype Based on Blockchain and Smart Contracts.* IEEE Internet of Things Journal, 2018: p. 1-1.
70. Uriarte, R.B. and R.D. Nicola, *Blockchain-Based Decentralized Cloud/Fog Solutions: Challenges, Opportunities, and Standards.* IEEE Communications Standards Magazine, 2018. **2**(3): p. 22-28.
71. Ibba, S., et al., *CitySense: blockchain-oriented smart cities*, in *Proceedings of the XP2017 Scientific Workshops*. 2017, ACM: Cologne, Germany. p. 1-5.
72. Fan, K., et al., *Blockchain-based Secure Time Protection Scheme in IoT.* IEEE Internet of Things Journal, 2018: p. 1-1.
73. Im, G.P. and R.L. Baskerville, *A longitudinal study of information system threat categories: the enduring problem of human error.* ACM SIGMIS Database, 2005. **36**(4): p. 68-79.
74. Spooner, D., et al. *Navigating the Insider Threat Tool Landscape: Low Cost Technical Solutions to Jump Start an Insider Threat Program*. in *2018 IEEE Security and Privacy Workshops (SPW)*. 2018.
75. Zhang, H., et al. *An Active Defense Model and Framework of Insider Threats Detection and Sense*. in *2009 Fifth International Conference on Information Assurance and Security*. 2009.
76. Yusop, Z.M. and J. Abawajy, *Analysis of insiders attack mitigation strategies.* Procedia-Social and Behavioral Sciences, 2014. **129**: p. 581-591.
77. Tukur, Y.M., D. Thakker, and I. Awan. *Ethereum Blockchain-Based Solution to Insider Threats on Perception Layer of IoT Systems*. in *2019 IEEE Global Conference on Internet of Things (GCIoT)*. 2019.
78. Tukur, Y.M. and Y.S. Ali. *Demonstrating the Effect of Insider Attacks on Perception Layer of Internet of Things (IoT) Systems*. in *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*. 2019.
79. Wüst, K. and A. Gervais. *Do you Need a Blockchain?* in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018.
80. Lyu, L., et al., *Fog-Empowered Anomaly Detection in Internet of Things using Hyperellipsoidal Clustering.* IEEE Internet of Things Journal, 2017. **PP**(99): p. 1-1.