Trust Management in Industrial Internet of Things using a Trusted E-Lithe Protocol

Ahmed Motmi[1], Samah Alhazmi[2]*, Ahmed Abu-Khadrah[3], Mousa AL-Akhras[4], Fuad Alhosban[5]
Computer Science Department, College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia[1, 2, 3, 4]
Computer Information Systems Department, King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan[4]
Computer Information Science Department, Faculty of Computer Information Systems, Higher Colleges of Technology, UAE[5]

***Abstract*—The IoT has gained significant recognition from research and industrial communities over the last decade. The concept of Industrial IoT (IIoT) has emerged to improve industrial processes and reduce downtime or breach in secure communication. If automated, industrial applications can make the implementation process more convenient, it also helps increase productivity, but an external attacker may cause distortion to the process, which could cause much damage. Thus, a trust management technique is proposed for securing IIoT. The transition of the Internet to IoT and for industrial applications to IIoT leads to numerous changes in the communication processes. This transition was initiated by wireless sensor networks that have unattended wireless topologies and were comprised due to the nature of their resource-constrained nodes. In order to protect the sensitivity of transmitted information, the security protocol uses the Datagram Transport Layer Security (DTLS) mandated by Secure Constrained Application Protocol (CoAP). However, DTLS was designed for powerful devices and needed strong support for industrial applications connected through high-bandwidth links. In the proposed trust management system, machine learning algorithms are used with an elastic slide window to handle bigger data and reduce the strain of massive communication. The proposed method detected on and off attacks on nodes, malicious nodes, healthy nodes, and broken nodes. This identification is necessary to check if a particular node could be trusted or not. The proposed technique successfully predicted 97% of nodes' behavior faster than other machine learning algorithms.**

***Keywords*—IoT; industrial; IIoT; trust management; E-lithe; secure communication; internet of things; CoAP; datagram transport layer security**

I. Introduction

Humans have been analyzing their surrounding physical environment for a thousand years, including identifying additional vital elements for maintaining the balance like measuring temperature, distance, and time. Initially, rudimentary methods were focused on using references like sun's position and body part sizes. With the technological advancement, the measurement units were standardized over time, the first-ever mechanical unit to offer exact physical measures appeared and were named sensors. With the electronic revolution in the silicon age, the method has been more precise for calculating and labeled as electronic sensors. United States Army introduced communication via sensors in the 1950s [1]. The research was initiated by Silverstein and was known as Sound Surveillance System. It was an intelligence-based project launched for detecting Soviet Submarines in the Pacific and Atlantic Oceans [2].

With time these sensors were improved for using trust-based communication through wireless network systems. The physical variables can now be modified through an artifact known as an actuator. The Wireless Sensor Networks (WSNs) are optimized thoroughly by incorporating sophisticated mechanisms and actuators, developing into Wireless Sensor and Actuator Network. Each node of the Wireless Sensor and actuator network can be turned into an Internet of Things (IoT) device using internet protocol [3].

IoT is conceptualized in this research as "An IoT device is an embedded system which is resource-constrained but has the capability of performing well-defined tasks like networking, signal processing, and sensing. It is powered by batteries and offers wireless communication capabilities" [2].

The concept of IoT has maximized the interoperability of devices. The connection and communication between devices have been facilitated but securing the connection is still questionable. The proposal for implementing IoT in computer connection and big data calculations is not new; however, the scope of the problem changes when implementing trust factors within the communication of Industrial IoT (IIoT) [4].

IoT is used for improvising domestic applications and has also been focused on innovating industrial applications. The research focuses on cyber security and trusted communication between industrial applications. Though IIoT offers quality domestic application uses, complex structures are needed to implement advanced communication techniques within industrial applications.

IIoT is carried out between hundreds of devices among hundreds or possibly thousands of devices connected to the same wireless network. It can create scalability issues, and an even larger amount of data transferred needs security and safe transmission without data theft and intrusion [5, 6]. Thus, the characteristic of interoperability needs to be controlled through improvising the security feature within IIoT. The efficiency of this technique may seem questionable considering implementing security features for a massive network that needs to be executed and maintained. Along with the

*Corresponding Author.

robustness and scalability requirements, the focus needs to be given to deploying fine-grained access control mechanisms [7].

According to Cisco and Gartner, currently, there are six billion users connected with the IoT devices; this number is increasing exponentially. It has been claimed by [7] that it is expected that IoT for industrial applications will help in improving security features and will offer great potential for research in this field in the coming years.

However, IoT has been associated with issues like resource-constraint devices with limited processing capabilities and their memory. The overhead is considered a technological barrier in this domain. The use of standard protocols only increased overhead delays and energy consumption [8]. The delays break communication connectivity, and overhead delays affect energy life. Both these factors are not acceptable for the efficient operation of any application. Thus, due to these issues, the questions stated below are developed to help direct research for bringing better prospects in deploying IoT in industrial applications and maintaining the trust factor [7].

1) Is the implementation of trusted communication for the Industrial Internet of Things feasible?

a) Are there any benefits of making IoT the baseline technology for IIoT Trust management?

b) If the performance impact is reduced, is it possible to increase interoperability?

c) Interoperability is the desired application of these services but increasing the possible number of inter-connections raises the chances of connecting many malicious users.

2) While maintaining performance, how can access exposed IoT nodes be controlled?

3) How can trust management be implemented while maintaining zero-configuration achieved for an IoT node?

To be able to answer these questions, a detailed analysis over energy, time consumption, and implementing trust factor in IIoT along with studying memory footprints and communication overheads, detailed analysis is conducted in coming sections of this research with the help of milestones depicted in Fig. 1 [7].

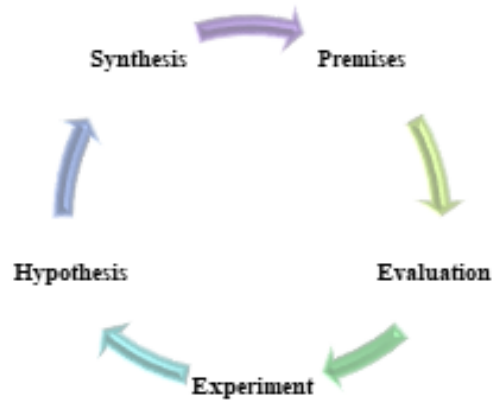

Fig. 1. Research Methodology [7].

The methodology described for this research is based on an iterative process depicted in Fig. 1. The iterative process needs a real-life problem. When the perimeters are targeted for the problems, it leads to the formulation of initial research questions. With the increasing knowledge, the depth of the problem may be well understood and lead to additional research questions. The iteration is expected to continue until the evaluation is successfully achieved [9].

The research will be divided into three stages. The first step is to examine the available research on network security, IoT technologies, Wireless Network Systems and authentication for IIoT, and the security protocols for lightweight key management through the preliminary study and literature survey. The second phase will be the theoretical design for resource-constrained sensor nodes in WSN and IoT using lightweight security solutions. The primary aspects of these proposed solutions were end-to-end (E2E) key management, device authentication, and communication in secure groups. Simulations, estimations, and real-time implementations in the third phase will evaluate the proposed solutions [10]. This research is aimed to evaluate through qualitative and quantitative research methods. Data will be collected from primary and secondary sources.

IoT requires collaboration from different research backgrounds and industries as a multi-disciplinary field. As industries are the major resource-generating entities, they need continuous improvement with evolving technology due to more human reliance and being operated manually; industries have been unable to operate optimally. This research proposes that IoT collaborate with industrial applications to produce an efficient IIoT model. All devices are expected to be automated and communicate through the Internet.

This research investigates, proposes, and analyzes efficient IoT technologies that will enable cutting-edge IoT networks to implement and update the existing designs into IIoT with the help of Wireless Sensors and Actuator Networks. This research focuses on securing communication and confidence among the nodes for industrial applications [11]. It is aimed to improve issues like scalability, security, dependability, energy efficiency, and interoperability which are expected to collaborate with the industrial applications and will help in providing a secure and efficient means for communication. In this research, a trust management model is proposed over IIoT by using an energy-efficient access control scheme. Additionally, this research aims at improving the existent IoT model energy and delays constraints.

The rest of this paper is organized as follows: Section II covers the Internet of Things, security protocols are covered in Section III. Methods and materials are covered in Section IV. Section V describes the method used to manage trust and detect attacks in IIoT. Section VI covers the results and discussion. Conclusions and possibilities for future work are covered in Section VII.

II. Internet of Things

For years, the Internet of Things (IoT) has been the technology of interest for innovating numerous other technologies. Innovation in industrial applications was

emphasized to ensure that technological advancement has been facilitated with the latest technology. However, issues like energy efficiency, delays, security, and safe communication are the main hindrances that need to be improved to successfully implement Trust management in IIoT [12].

IoT is a vast topic that cannot be defined in a single definition. A possible definition of IoT is a collection of services that connect objects, whether electrical, electronic, or non-electrical, to offer contextual services and seamless communication. Development of services like mobile phones, actuators, sensors, and Radio Frequency Identification (RFID) tags can help assess IoT's facility to facilitate human needs. It is also known as a network of embedded sensors for increasing the ease of connectivity. IoT also means 'internalization of every connected object. It allows humans to control the means of communication and data transfer. Even if the objects are modified, or technology is enhanced, there are possible chances that human involvement will not be required.

In this research, the following definition of IoT will be adopted: A device connected to an IoT network is a resource-constrained embedded system with the ability to perform multiple tasks simultaneously like signal processing, networking, and sensing. Batteries usually power it to consume lesser power and provide wireless communication capabilities.

The concept of IoT may change or evolve due to changing software and hardware technologies or the need for industrial applications to evolve. Implementing IoT in industrial applications will help improve efficiency and offer trusted communication among devices as IoT faces issues like data hacking or external intrusion [13]. If this happens with any industrial application, there are chances that it might damage a device or make it vulnerable to data theft. This research investigates how secure and trusted communication will be implemented in IIoT.

The concept of IoT involves numerous software components; however, the latest evolution has been made in link layer and application layer protocols and operating systems. Recent advances in the IoT domain, such as the Application Protocols**,** the OSI model represents that the application layer is known to be the abstraction layer known for acting as an interface between the applications running on the host and how it is communicating with the user.

The following list includes known application layer protocols for IoT [14].

- RESTful HTTP is the first IoT protocol acknowledged for executing Hypertext Transfer Protocol (HTTP). It is mainly used for web-based services in which most of the work is to facilitate communication between the client and the user. The transport layer is deployed in the TCP protocol. However, the usage of XML makes it inefficient for low-power purposes and complex for general usage. The latest improvements made in HTTP have enabled the header compression to improve the overall performance of the HTTP protocol. The overall power consumption issue has been suitably dealt with, but it is still inefficient for implementation in a resource-constrained device like IoT [15, 16].
- MQTT: based on a client broker-server architecture, the MQ Telemetry Transport protocol created by IBM is implemented using two types of communication processes, i.e., Publish/Subscribe and in HTTP as Request/Response. This protocol still uses TCP, but it is more efficient than HTTP.
- Jabber: this protocol was developed by an open-source community to support instant messaging. Similar to MQTT, communication depends upon XML. It supports the client-server model using both communications mediums, i.e., Request/Response and Publish/Subscribe. However, this protocol also uses TCP in the transport layer [17].
- XMPP: Jabber protocol was modified by Internet Engineering Task Force (IETF) by including SASL for authentication and TLS for communication encryption. It is supported in extensible messaging and presence protocol.
- MQTT-SN: IBM proposed a modified UDP-based version of MQTT, which is more efficient and used in Sensor networks.
- Web-Sockets: This protocol was designed to improve communication between web servers and browsers; however, apart from these services, it can be used independently as a client-server application protocol. This protocol also relies on TCP for the Transport layer.
- CoAP: The Constrained Application Protocol (CoAP) was developed for optimizing the efficiency of communication in WSN. This protocol, known as the Restful-based protocol, has been enabled to execute its services directly on network nodes. Depending upon the client-server model, it observes methods, and depending upon these methods; it allows the Request/Response procedure. Unlike other protocols, this protocol uses UDP protocol instead of TCP protocol in the Transport layer [18].

Link-layer protocols: The innovation of wireless technologies like Bluetooth and Wi-Fi has introduced Wireless Local Area Networks (WLANs). It has served as an optimal technique for every mobile sensing platform and a gateway for both techniques. However, the only barrier faced using this technology is power consumption. The device's battery dies down within a lesser time when these techniques are not used. Thus, it lessens the time for consumption of Bluetooth and Wi-Fi in any device. Numerous new improvements have been made in hardware components to reduce power consumption, like the one manufactured by a Texas instrument named CC3000. It has a reception consumption of 331 mW and a transmission consumption of 936 mW. The wireless technology consumes the majority of the power of any IoT device; consequently, while selecting the wireless technology for the device, it shall be observed how it affects the power. The device can operate for an extended period without exhausting its batteries. The IETF in 2006 developed a link-layer protocol 6LoWPAN with header compression and encapsulation. The primary purpose behind it was to use IPV6

networks. It was the most significant innovation for creating IP wireless networks for low-power devices [19].

Since the past decade, the use of embedded systems in industrial applications and other evolving technologies like upgrading cell phones has emphasized the innovative development of these systems. It can be seen currently that smart homes, safe cities, and automated gates result from these embedded systems. For this purpose, the hardware has been improvised, which will be discussed in this section.

The innovation of microprocessors and microcontrollers proposed the latest smaller technique and used lesser computational power. Previously, the IoT devices have been using microcontrollers due to lower computation power; however, with the current need to implement IoT in industrial applications for better and safe means of communication, the need for Microprocessors was felt. Intel Atom and the ARM Cortex-M73 result from this innovation, which consumes less power and is high-performance. Microprocessors are recommended for nodes that require a high level of power in the processing and mitigate the overhead in the communication [3].

The CoAP services are located in the link-layer and are used to design web-based services capable of working with resource-constrained devices. This technique is efficient for microcontrollers that can run over 6LoWPAN network stacks and have a small ROM and RAM. However, it gives high error rates in a packet transfer. The devices using this technology can switch to sleep mode to save power and give optimal performance for low-power networking. The Request/Response interaction model is provided by CoAP between the communication ends of the applications. This protocol supports key Web concepts, built-in discovery, extensible header options, and RESTful interactions. For integration with the web, CoAP can easily develop an interface with HTTP; it will help fulfill the needs of constrained environments like very low overhead, multicast support, and simplicity of procedures. The features of CoAP which are relevant to the research are discussed below:

- Two types of messages are transmitted, a confirming message with the exponential expiry time to receive the acknowledged message. On the other hand, a validation message is sent without the expected response from the server.
- Uniform Resource Identifier (URI) format uses specialized service endpoints and standard services. One example explaining the procedure is /.well-known/core path in RFC 5785, and the name Core Format knows another format.
- CoAP can send large messages in blocks with the stop and wait for mechanism. In this way, no data packet will be lost, and the complete message will be transmitted in 'Block wise transfers' [4]. Fig. 2 illustrates the transfer of message block-wise.

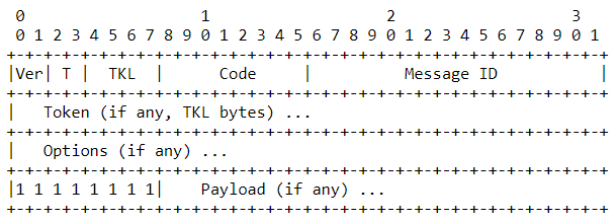

Fig. 2. CoAP Packet Format [3].

Providing E2E security is a widely discussed topic in conventional communication using the Internet. Though E2E was explored a long time ago, little research is available on E2E security using 6LoWPANS. The lossy nature of wireless links and the device's resource constraints are the main reasons for not applying E2E security mechanisms details to 6LoWPANs. IP-based IoT faces security challenges during the handshake process. To resolve security issues, it is suggested to: (1) validate certificates at the trusted 6BR, (2) a full handshake shall be avoided by session resumption, and (3) the owner of the resource-constrained device shall allow the handshake procedure. The certificate-based authentication is feasible for this type of authentication [4].

Due to heterogeneity in IoT, it becomes difficult to connect resource-constrained devices in a more secure and reliable way. Especially when it comes to the connection in industrial applications, apart from operating in a resource-constrained environment, the ruggedness of the environment and weather have to be considered. To enable this process to be implemented in industrial applications to ensure trusted communication between nodes, the IETF has proposed techniques using existing protocols like CoAP, the IPv6 Routing Protocol, and 6LoWPAN. These protocols proved to be useful for lossy and low-power networks. The Datagram Transport Layer Security (DTLS) protocol guarantees E2E security for different applications running on the same machine. It operates between the application layer and the transport layer. The DTLS consists of two layers; the upper layer includes any three stated protocols like application data, ChangeCipherSpec, Handshake, and alert.

The ChangeCipherSpec indicates that the Record protocol should protect the messages with security keys and a newly negotiated cipher suite during the handshake procedure. DTLS uses the alert protocol for communicating error messages due to lossy networks within the DTLS layers. Once the handshake procedure is completed, the record header is mainly responsible for cryptographically protecting the application data or upper-layer protocols. The record protection protocol offers authenticity, integrity protection, and confidentiality features. The handshake procedure is a chattier procedure than a DTLS protocol as it releases numerous messages in a synchronized fashion [4].

III. Security Protocols

A. Wireless Personal Area Networks 6LoWPAN

Fragmentation and header compression mechanisms of IPv6 datagrams are defined within the 6LoWPAN standard. The IPv6-connected WSNs are also known as IPV6 networks. The compression mechanism used in this protocol is Next

Header Compression and IP Header Compression (IPHC). Like IPHC, DTLS header compression can be applied between 6BR and sensor nodes within the 6LoWPAN networks. The complete information required for routing has been extracted from the IP layer. It happens because DTLS headers are part of the payload scheme. 6LoWPAN header compression mechanisms compress the headers in a UDP payload. To perform 6LoWPAN compression, a new modification is required in which a new NHC for UDP with different ID bits is assigned. This approach will extend the existing 6LoWPAN and be easier to implement than making changes to the existing technique [20]. Table I shows the reviewed protocols.

TABLE I. REVIEWED PROTOCOLS

Year	***Protocol***
1999	MQTT client broker-server architecture [21]
1999	Jeremie Miller announces the existence of Jabber [22]
2000	Roy Fielding first presented RESTful [21]
2004	Publishing XMPP standards: a modified version of Jabber protocol [23]
2011	WebSocket improved computer communications protocol [24]
2012	Trust-based communication – WSNs [25]
2012	DTLS a protocol that guarantees the implementation of E2E security [26]
2014	Constrained Application Protocol (CoAP) [27]
2017	6LoWPAN an approach for routing IPv6 over low-power wireless networks [28]
2017	Enhanced Lightweight DTLS for IoT [29, 30]

B. E-Lithe

IoT made the connection of millions of devices possible. However, developing secure communication is a challenge for IoT devices. If secure and trusted communication between Industrial applications using IoT is not possible, it does not only threaten productivity and efficiency, but it also threatens important data used within these industrial applications. It is proposed to provide secure communication within the IoT environment by implementing DTLS while constructing a secure transport layer over the datagram. DTLS is a protocol that is expected to provide secure communication in client-server applications. The mechanism depends upon transport layer security which prevents fragmentation, tampering, and message forgery. This protocol also deals with the datagram's size, loss of datagram, and packet re-ordering. However, an issue is identified that the DTLS protocol is defenseless against the Denial-of-Service (DoS) attacks [31] and requires more computation than an average device operation while working in a resource-constrained device. DoS attacks prevent the communication between two nodes and can disrupt the network services, thus, disrupting the communication between complete networks. DoS attack is identified when the requested services are not provided to the user due to an attack on one of the networking devices [8].

To overcome DTLS shortcomings for constrained devices, an Enhanced and lightweight DTLS protocol was proposed and named Enhanced Lightweight DTLS for IoT (E-Lithe). For this research, a trusted third part element will be added to E-Lithe for implementing E-Lithe in IIoT to manage trusted communication. The trusted third-party feature aimed to prevent the DoS attack by pre-sharing the secret keys. The E-Lithe protocol is explained as below:

- The server and the third-party trusted protocol agree on sharing a secret key before beginning the handshake procedure.
- A mutual secret key is shared between the client and a Trusted Third Party.
- The sharing of the mutual key prevents the power exhaustion of devices and authenticates the client-server communication.
- The client sends a handshake message to the server.
- If the server confirms the validity of the key, the server generates a hello message in return for the client response. However, if the keys are not matched, the process is terminated.

E-lithe uses lesser power during message transmission to prevent the overloading of fragmentation by applying the compression technique to ensure the lightweight transmission of messages. The compression strategy used for E-Lithe comprises a client Hello, a handshake layer, and a record layer. On the other hand, the record layer comprises a fragment, a sequence number, and an epoch. The handshake layer consists of message sequence and message type. The message is sent precisely with message type, and length details are ignored. Fig. 3 depicts the communication mechanism in E-Lithe.

C. Distributed Trust Management System

The devices connected through IoT face the issue of secure communication. Insecure communication could bring more devastating damage if it happens in industrial applications. It does not only damage the device, but it can also give access to sensitive data. The main issue is to identify malicious attacks before the handshake procedure. These malicious nodes choose selective attacks which require lesser processing requirements.

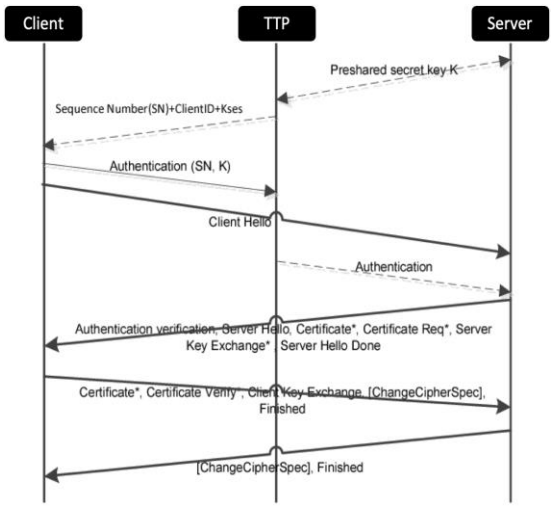

Fig. 3. Communication in the E-Lithe Scheme [8].

Trust management will check the fault in the network and focus on protecting nodes and networking connectivity. The main scheme for trust management is to implement trust among the connected devices. While checking the trust management, they also focus on checking the behavior of malicious nodes. In IoT, several trust management schemes are available, like centralized, decentralized, and hybrid.

The trust management value of each node is calculated on direct observation of the nearest nodes. This value is zero at the start. The start value shows no trust between any two devices at the beginning as trust needs to be built. An announcement is sent to the nearby nodes, and the value is calculated through the sending nodes. The service provided by nodes is denoted as a healthy node and is considered a broken node if the service is not provided on time.

A policy-based secure and trustworthy sensing scheme called "Real Alert" has been observed for this research. In the mechanism, IoT node attributes and the data trustworthiness are calculated through evaluating the anomalous data and the contextual information from which this data is collected. The monitored direct trust value measured from network communication relies on the quantitative value of the trust model. The evaluated features are integrity and delay, consistency of the packet content, repetition rate, and packet forwarding capacity. The D-S theory is computed to calculate trust. However, the drawback of this scheme is that it uses a large amount of data, and streaming this amount of data creates a problem for the conventional networking system.

The redemption scheme and the trust management differentiate between malicious behaviors to detect and defend against On and Off attacks and temporary errors. The ratio of good behavior to the total behavior is calculated using the difference as the predictability trust, and a static sliding window is used for recording previous behavior [32].

D. Naïve's Bayes Theorem

Machine learning algorithms are proposed to be an alternative for calculating trust values among the connected nodes. This theorem can help from attempting multiple calculations. The BAN-Trust scheme will have opted for this scheme on the recommendation of other nodes detecting an off behavior of any node. In this process, if any node identifies and reports that the other particular node is behaving off or sending flooding messages, the system can ban that node from the communication network. In this way, the rest of the network is protected [33, 34].

A Naïve's Bayes trust management model is easier to build and is less complex. It can use large datasets and does not rely on iterative parameter estimation. The Bayesian theorem is stated as below:

$$P(x|y) = \frac{P(x|y) \times P(x)}{P(y)} \quad (1)$$

Where $(x|y)$ = probability of a class, x given instance, y, $p(y|x)$= probability of instance, y given class and x, $p(x)$=probability of occurrence of class x, $p(y)$=probability of instance y occurring.

The Bayesian theorem uses one parameter only. This parameter calculates all features and simplifies them through the Naïve Bayes theorem for numerous features. This theorem can be used for detecting malicious nodes. The features of the Naïve Bayes classifier are Packet Loss Rate and the Packet Error rate. Malicious nodes intentionally disseminate erroneous packets or drop packets. For this purpose, the packet loss rate and the packet error rate are included for calculating the trust value between nodes. According to the Bayesian theorem, trustworthiness can be classified as High, Low, or Moderate [33]. The calculation of the Level of Trust in Naïve Bayes is explained in Fig. 4.

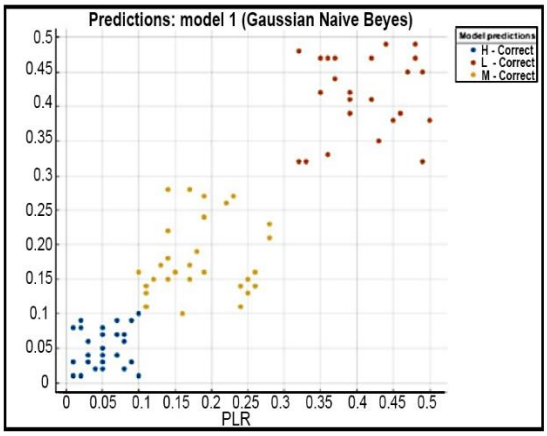

Fig. 4. Classification of Trust Values after Training the Model.

IV. Methods and Materials

To present the real-time implementation of the improvised E-Lithe, Contiki is used. It is an open-source operating system used for implementing IoT. The proposed header compression technique can be implemented with the support of 6LoWPAN. Cooja enables cross-level simulations at many levels and supports the 6LoWPAN protocol with a more convenient interface. Cooja can combine low-level and high-level simulations of sensor node hardware and how they behave in a single simulation. The Cooja simulator provided a flexible approach for implementing improved E-Lithe for trust management in IIoT. The partial real-time scenario generated with the help of Cooja will give a better insight for deploying this research at the industrial level. The improved lithe implementation requires the support of four components: CoAP, DTLS, DTLS header compression, and the CoAP-DTLS integration module. Open-source Ubuntu 14.04 LTS 64bit will be used for DTLS implementation. It uses the pre-shared keys: TLS_PSK_WITH_AES_128_CMC_8 for supporting the basic cipher suite. For the WiSMote Platform, Ubuntu 14.04 LTS 64bit and VM Ware workstation are used. The default CoAP implementation will be used for CoAP implementation in Conitki. An integration module will create a collaboration between DTLS and CoAP and enable the CoAP protocol. Independent application access is created due to this integration with CoAP. In this process, the CoAP messages are handed over to DTLS, responsible for transmitting them to the receiver's end [35].

Initially, all the CoAP messages are received at DTLS. Once processed and checked, DTLS transfers these messages to CoAP, stored at the application layer. The header compression will be used, as an extension, to implement 6LoWPAN in Contiki. The 6LoWPAN layer has been placed between the Medium Access Control and IP layers. The packets ready to be transmitted from the nodes in the IP layer are known as Output packets. The packet received at the node from the MAC layer is known as the input packets. However, the 6LoWPAN layer can process the UDP packets from both directions. The UDP packets depending upon the messages, are divided into two categories. The default DTLS port for pre-configured input packets identifies CoAP messages. In addition to it, the security shall not be compromised during the E2E sharing of keys during the header compression scheme. Yassine's is a secure version of DTLS [35]. The Yassine's version of DTLS and E-Lithe are observed to have closer values, as shown in Fig. 5. However, when compared and shown results through the graph depicted in Fig. 5, it has been identified that the E-Lithe can handle the compressed data packets better than the heavy data packets. E-Lithe uses the header compression scheme despite the cookie exchange scheme, which helps it perform better.

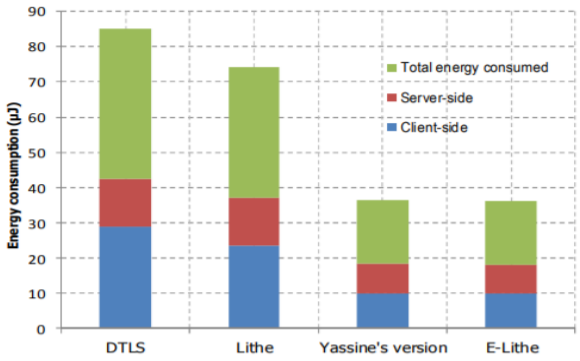

Fig. 5. Total Energy Consumed for Each DTLS Variant.

Improved Lithe will be evaluated by implementing sensor nodes in Contiki. As a hardware platform, WiSMote will be used. WiSMote proposes the features like a 16-bit RISC microcontroller, MSP430 5-Series, 16 MHz, an IEEE 802.15.4 (CC2520) transceiver, and 128/16 kB of ROM/RAM. WiSMote was selected due to the RAM and ROM requirements of DTLS.

The network will consist of forty WiSMote. One of the WiSMote will serve as a server that will communicate with other nodes. As this research aims to identify the broken nodes and the attacked nodes, these nodes will be sending *Hello Flood* Messages to other nodes. The attacked nodes are essential to be identified because they are the loop through which data theft can happen, or any foreign intrusion is expected. The communication between nodes that are attacked or broken will be restricted. They will not be able to send messages. The rest of the nodes performing on time can be called trusted nodes. It is essential to ensure that the nodes within the network are trusted because a single broken or attacked node within an industrial network can damage the complete network and may cost resources for replacing the hardware or cause downtime.

V. A Smart Trust Management Method to Detect On-Off Attacks in the IIoT Applications

The proposed approach in this research aims to detect on and off attacks and broken nodes on networks for IIoT. The communication between the industrial applications will be calculated through the available metadata attributes. IIoT metadata can be evaluated by sending it to the proposed algorithm.

Data is entered into the feature type extraction process for the pre-processing phase. Hashing vectorizer is used for processing text data. The text in this format is converted into token occurrences of a matrix. The integer index mapping string is named by the token string name. This approach is used because it does not need the support of the dictionary and can be used for streaming. The pre-processed dataset is fed to a machine learning classifier for identifying the class. Few limitations have been accepted for evaluating the industrial data, such as calculating the average temperature for a city. If the temperature is within range, it will be called trusted data, but if it is out of range, it can be labeled as a broken or attacked response, although it could result from extreme, unusual weather conditions. A decision function value is returned if a classifier confirms an identified class. The methods adopted the use of the decision function for calculating the size of the Elastic Slide Window. It is evaluated by observing the model decision function of the distance hyperplane of the sample data. A high positive decision value is received corresponds to high prediction assurance, as illustrated in Fig. 6 and Fig. 7 [4].

Fig. 6. Expected Range of Trusted Value [4].

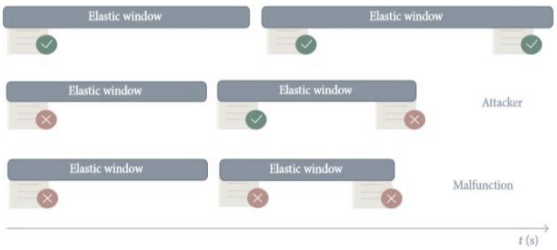

Fig. 7. The Elastic Slide Window [4].

The introduced Elastic Slide Window is an essential concept for data flow. Using the time frame analysis, it enhances trust. All the values are either good or bad during the On and Off attack are sent in a discretionary manner. If the system is healthy, it will accept the good value over time, but the suspect of being an attack is always expected. If the identifier sends a low decision function value or an identified class, the trust for that value is doubted, and that particular resource is either expected to be tested again for trust, or its communication is limited. The decision function values evaluate the size of the elastic slide windows. Any time a low

decision function value is received, the elastic slide window is increased for evaluating the trust of that particular decision function. The trust dispatcher records the storage size of the elastic slide window in the database. This feature also determines the trust response, i.e., Good, Broken, or on and off attacker. The proposed algorithm is illustrated in Fig. 8.

```
input: A metadata m (ID, read)
output: A predicted type: (Tusted, On-OffAttacker, Broken)
(1)  eswAlpha ← alpha;
(2)  eswInit ← beta;
(3)  NewPrediction ← Classifier.Predict(m);
(4)  NewDecisionFunction ← Classifier.DecisionFunction(m);
(5)  if m in Database then
(6)      m.SlideWindow ←
           (eswInit + time()) − NewDecisionFunction;
(7)      m.prediction ← NewPrediction;
(8)      if m.SlideWindow ≥ Time() then
(9)          if NewPrediction == −1 and m.prediction == −1 and
               NewDecisionFunction ≤ eswAlpha then
(10)             m.prediction ← 0;
(11)         end
(12)         if NewPrediction != m.prediction and
               NewDecisionFunction ≥ eswAlpha then
(13)             m.prediction ← −1;
(14)         end
(15)         if NewPrediction != m.prediction and
               NewDecisionFunction ≤ eswAlpha then
(16)             m.prediction ← m.prediction;
(17)         end
(18)     end
(19) end
(20) m.SlideWindow ← m.SlideWindow − NewDecisionFunction;
```

Fig. 8. The Smart Trust Management Algorithm for Industrial Internet of Things.

The smart trust management server will consult, via the Constrained Application Protocol, for an object, and the value is returned in JSON formatted data. Honesty, exploitation, and selfishness levels determine the node's trust. The node will be tagged as an attacker node if the trust value is not satisfied with the threshold value. For example, the object Id: 15 with a metadata payload of 45 degrees Celsius is marked as an On and Off attacker. The decision function for a trust score value is also presented. Other expected results can be transcribed as Good for predictable devices or even two broken nodes in the same elastic slide window.

VI. Results and Discussion

This research's proposed trust management method is expected to detect On and Off attacks in IIoT with 97.1% precision tested on a real-time dataset. For the simulated environment, 95% of precision is achieved. The reasons behind choosing the proposed method were: (1) to establish secure communication with resources limitation presence in IIoT, (2) to enhance the defenseless DTLS protocol against DoS attacks to gain the advantage of lightweight transmission with low power consumption. Compared to other studies, the proposed method is 95% faster, and On and Off attack is predicted 5% more accurate in On and Off identification attacks. The Elastic Slide Window feature helped identify the malfunctioning or broken nodes among the misbehaving devices by evaluating the exploitation, selfishness, and dishonesty levels. Fig. 9 and Table II show the simulated node behavior based on delay and overhead.

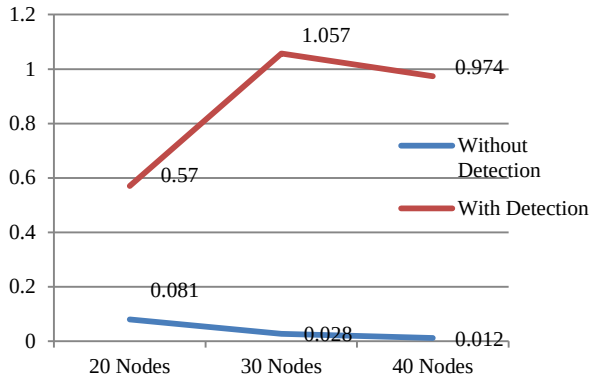

Fig. 9. Power Consumption in Nodes vs. Delay.

TABLE II. The Relationship between the Number of Nodes and Delay Time with and without Detection

Number of Nodes	*Without Detection*	*With Detection*
20	0.081	0.57
30	0.028	1.057
40	0.012	0.947

The decision function boundaries are created by joining the trained datasets, which are blue in the illustration. The x-axis in Fig. 9 represents the number of nodes, whereas y-axis represents delay time for communication. The OneClassSVM classifier efficiently grouped the test data, which was near to the trained dataset. The abnormal samples opted in this research were presented far from the decision function values. Typical or trusted values are near 0 for an identified class. The broken, attacked, or misbehaving nodes give high distance values up to -200 [3].

The proposed method could be affected by suspected validation threats like datasets, classifiers seeds, and random simulation outputs. The output of each simulation varies according to its run time. To minimize the error in the validation threat, the simulation is executed three times, and the average of these three simulations is used as a result. During the training tasks of the models, scikit-learn library procedures create an alert for the users that they use parameter initialization variable and random seed values, which together contribute towards different types of precision values in results. The average values were annotated for three fitting rounds for the simulation values. The null values were sanitized from the real-world dataset by removing the non-related data and NaN values. However, the related work sessions could be threatened if no relevant study is considered during the concept development. To minimize the risk, main indexing databases were considered for verifying references and citations of sources within the context to validate the theories.

The previous sections discuss the improvisation in the most well-known technique, 'Lithe,' to implement security and trust-based communication in IIoT. The selected trust management technique is analyzed and evaluated, relying on essential trust metrics: scalability, availability, adaptability, reliability, privacy, integrity, and accuracy. The selected articles show that the researchers have focused on implementing security and no

data loss communication between nodes. While researching the available data, it has been identified that there is limited data available in research regarding Trust management in IIoT. The security feature could be established for a smaller network. However, the industrial network is massive. It needs to be protected and ensured by implementing improved E-Lithe to prevent DoS attacks on devices and establish that the devices within the network could be trusted because they have been responding within range and within the expected time.

Table III illustrates that the researchers have focused on a few parameters for conducting research like scalability, adaptability, availability, accuracy, and security. However, the features like establishing the fact that the nodes in the network are not broken, attacked, or misbehaving nodes, which can be dangerous for the complete network and its authenticity, are not discussed. The attacks not only threaten the safety and secure transmission of sensitive data but also damage the network device.

TABLE III. PARAMETERS FOR SIMULATION

Network Simulator Parameters		
Parameter	***Value/Description***	***Remarks***
Number of nodes	40 nodes	255 for each scheme
Simulation area	1000 x 1000 m	Controlled by wireless coverage
Topology	Random	Determined by Cooja (Simulator)
Radio medium	UDGM	Directed Graph Radio Medium
Routing protocol	RPL	IPv6 Routing Protocol for Low power and Lossy Networks
Mote type	WiSMote	Contiki-wismote-platform
Packet analyzer	Wireshark	Network protocol analyzer
Packet interval	10 seconds	10 ms to 60 S

Due to the fast growth of the technology and the network requirements, the IoT network cannot handle the trustworthiness computation. The implementation of Lithe ensured the secure communication of data, but this technique does not support the trusted communication of data between devices. The technique faces the issues like fake recommendations from other devices. A hacked device can send messages for other devices that they are hacked, limit their communication, downgrade the ranking, which may not be part of the system, or upgrade the rank of any device once they are fit for use. Due to this, they may predict the wrong trust value.

The trust decision is developed on static rules, whereas the trust decision must be taken dynamically during trust negotiations. Due to its constrained environment, the improved E-lithe cannot be directly applied in the IoT. The devices need to be upgraded at the application and the transport layers to prevent loss in the data packets. Even if the message is paused, it does not take much time. The message reaches its destination in a complete form.

In this section, Cooja's implementation will be discussed. Initially, the installation of the software will be explained. For implementing the trusted communication between IIoT, Contiki 3.0 open-source OS is used with supporting software of VMware. The aim was to generate simulation with the pre-processed data for identifying how nodes can communicate by managing the trust factor. It is a partial real time scenario. All the variations will be tested here so that once the program is fully completed, it is launched easily for the real-time industry without the fear of failing. Studying traffic volume, position, and number of malicious nodes at least requires 120 minutes for determining the On and Off attacks. However, the method proposed is expected to detect the On and Off attacks in approximately 10 minutes which is 95% faster than other methods. The method offers a faster way for prediction and is 96% more accurate. Node31* (Good), 8* and 32* are (attackers); Nodes 5, 9, 22 are (Good) and 12, 13 and 15 are on and off attacked, therefore they cannot be trusted. It is generated from the simulated scenario as shown in Fig. 10.

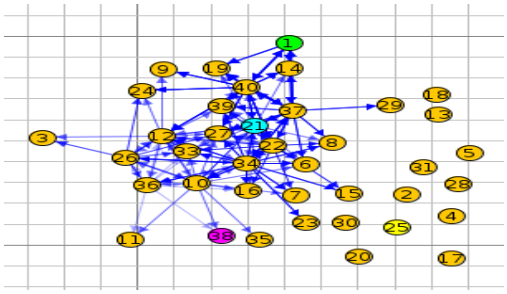

Fig. 10. DODAG Attacks Generated in Cooja Contiki 3.0.

Node*5 is identified as a selfish node and was identified in the first seven minutes of simulation execution. The positive trust score is related to good nodes, which could be trusted, and negative values are related to attacked nodes.

The above scenario is one of the probable situations in which not all the nodes are attacking nodes; they may be broken or selfish, and due to this, they have been unable to send messages within range. However, and to the best of our knowledge, there is no relevant research available regarding the message sent in reply that the particular node can be trusted or not and for broken nodes as well that the following node is broken and can participate. Thus, the trust-based mechanism for IIoT has been calculated depending on factors like Dishonesty, Exploitation, and Selfishness levels.

The dishonestly level depends upon the packet-dropping nature. If there is any packet loss during the communication process, the message will not be received in a complete form at the destination. It is pre-conditioned here that if the dishonesty level is greater than one, the node is suspected of being dishonest.

The exploitation level is calculated based on the over flooding of hello packets or Distributed Interactive Simulation (DIS) message. If Dishonesty Hack (EH) level is greater than 1, it is determined as a suspect.

Selfishness level is determined based on the attractive nature of the node. Typically root node alone had higher quality than other nodes. However, the attacker node proposed

a lower rank than the root node to attract the attackers. If the Selfishness hack (SH) level is greater than 1, it is determined as a suspect. Trust calculation:

$$Trust\ (Ti) = (Weight \times \text{Hi}) + (Weight \times \text{Ei}) + (Weight \times \text{Si}) \quad (2)$$

Where *Weight* factor = 0.33, H_i is Dishonesty level, E_i is Exploitation level, and S_i is Selfishness level. If trust is greater than 1, the node is determined as an attacker.

For validation purposes, the annotated dataset was found useful. The method was compared to other machine learning algorithms like linear SVM, Neural Net, Naïve Bayes, and K Neighbors Classifier. Comparisons are illustrated in Table IV.

The proposed method was able to identify two good nodes, two broken nodes, and three attacking nodes. Fig. 11 shows the types of nodes identified. The actual nodes are shown in the form of filled nodes. The color of nodes is depicted with names at the end of the image. The mark around the nodes presents the predicted class.

For developing and evaluating intelligent data middleware, we used 3111 samples of temperature data collected by 116 sensors from February to March 2019 for Arahnus in Denmark. The average temperature in Arahnus ranges from -3 to 16 degrees Celsius. A total of 501 misbehave samples were simulated using random out-of-range temperature observations. Fig. 12 presents the attacking dataset captured by Wireshark. It gave similar results to the simulation, and approximately 97% of precise results were generated.

TABLE IV. COMPARISON WITH SUPERVISED CLASSIFIERS

	Classifier				
	Linear SVM	***Naïve Bayes***	***Neural Net***	***Nearest Neighbors***	***Our Method***
Precision	0.88	0.92	0.92	0.91	0.96
Pecall	0.71	0.82	0.81	0.84	0.85
F1-Score	0.74	0.85	0.84	0.84	0.87

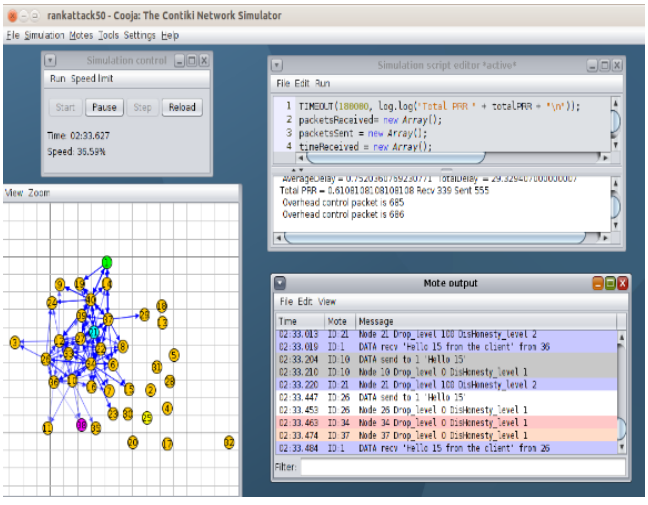

Fig. 11. Rank Attack for Nodes Cooja Contiki 3.0.

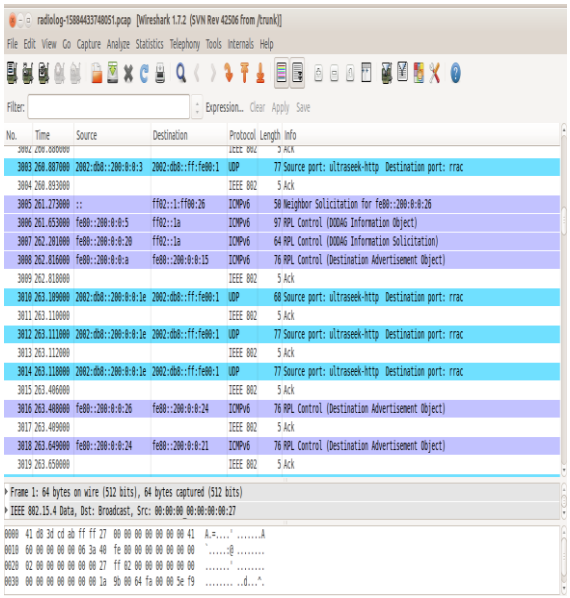

Fig. 12. Attacking Dataset Captured in Wireshark.

The output results from one-class classifiers types have been used in the proposed method. The OneClassSVMclassifier has been identified to help identify the attacking nodes and prove trusted communication between nodes. The one-class classifier is used to justify the aim that the trusted communication between the nodes has to be established by separating good nodes from the broken nodes and limiting the communication of malfunctioning nodes. However, it is hard to tag a classifier for each type of metadata.

VII. CONCLUSION AND FUTURE WORK

Implementing trust management in the Industrial Internet of Things (IIoT) is challenging. The relevant research available has been focused on implementing IIoT networks, but little focus has been made on IIoT trust among the nodes. Though this is a new concept, it is limited by the little data available. Using the current research available, a smart trust management system has been introduced using machine learning algorithms and size of Elastic slide window. The proposed method successfully detected On and Off attacks on nodes, malicious nodes, healthy nodes and broken nodes. It is necessary to identify the type of node within the network for identifying that if particular nodes could be trusted or not. The proposed technique was able to predict 97% of accurate behavior of nodes faster than other machine learning algorithms. The Elastic Slide window introduced has the capacity of identifying broken nodes, malfunctioning nodes or attacking nodes in industrial network as they have the capacity of handling bigger data and take the strain of massive communication.

This research examined the available research on network security, IoT technologies, Wireless Network Systems and authentication for IIoT, and the security protocols for lightweight key management through the preliminary study and literature survey. Then we proposed E2E key management using a lightweight security solution to achieve device authentication and communication in secure groups:

simulations, estimations, and real-time implementations evaluated and validated such solutions.

The proposed system can be tested against other machine learning techniques and scenarios in more complex industrial networks. Currently, only the trust factor is evaluated. If further research is conducted, there are possible chances of implementing an intranet network facility using E-Lithe approach for ensuring that the particular network can only be accessed by authorized people and can deal with any incoming attacks. Also a random dataset can be used to test if the mechanism is able to identify the validity and trust factor within the system. Also, elastic slide window can be used for identifying other IoT related trust based attacks in industrial environments like ballot-stuffing attacks, opportunistic service attacks, bad-mouthing attack and self-promotion attacks.

ACKNOWLEDGMENT

Thanks to the Saudi Electronic University for sponsoring this work.

REFERENCES

[1] S. Al-Rubaye, E. Kadhum, Q. Ni, and Anpalagan, A. "Industrial Internet of Things Driven by SDN Platform for Smart Grid Resiliency." *IEEE Internet Of Things Journal*, *6*(1), pp. 267-277, 2019. DOI: 10.1109/jiot.2017.2734903.

[2] G. Falco, C. Caldera, and H. Shrobe. "IIoT Cybersecurity Risk Modeling for SCADA Systems." *IEEE Internet Of Things Journal*, *5*(6), pp. 4486-4495, 2018. DOI: 10.1109/jiot.2018.2822842.

[3] M. Hasan, and H. Al-Rizzo. "Optimization of Sensor Deployment for Industrial Internet of Things Using a Multiswarm Algorithm." *IEEE Internet Of Things Journal*, *6*(6), pp. 10344-10362, 2019. DOI: 10.1109/jiot.2019.2938486.

[4] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie. "Towards Edge-Based Deep Learning in Industrial Internet of Things." *IEEE Internet Of Things Journal*, pp. 1-1, 2020. DOI: 10.1109/jiot.2019.2963635.

[5] X. Liu, H. Huang, F. Xiao, and Z. Ma. "A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs." *IEEE Internet Of Things Journal*, pp. 1-1, 2019. DOI: 10.1109/jiot.2019.2957421.

[6] A. Alyousef, K. Srinivasan, M. S. Alrahhal, M. Alshammari, and M. Al-Akhras, "Preserving Location Privacy in the IoT against Advanced Attacks using Deep Learning" International Journal of Advanced Computer Science and Applications (IJACSA), 13(1), 2022.

[7] http://dx.doi.org/10.14569/IJACSA.2022.0130152

[8] P. Ray, M. Mukherjee, and L. Shu. "Internet of Things for Disaster Management: State-of-the-Art and Prospects." *IEEE Access*, *5*, pp. 18818-18835, 2017. DOI: 10.1109/access.2017.2752174.

[9] J. Mcginthy, and A. Michaels. "Secure Industrial Internet of Things Critical Infrastructure Node Design." *IEEE Internet Of Things Journal*, *6*(5), pp. 8021-8037, 2019. DOI: 10.1109/jiot.2019.2903242.

[10] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe. "Blockchain-Enabled Decentralized Trust Management and Secure Usage Control of IoT Big Data." *IEEE Internet Of Things Journal*, pp. 1-1, 2020. DOI: 10.1109/jiot.2019.2960526.

[11] C. Zhu, J. Rodrigues, V. Leung, L. Shu, and L. Yang. "Trust-Based Communication for the Industrial Internet of Things." *IEEE Communications Magazine*, *56*(2), pp. 16-22, 2018. DOI: 10.1109/mcom.2018.1700592.

[12] H. Tschofenig, and E. Baccelli. "Cyberphysical Security for the Masses: A Survey of the Internet Protocol Suite for Internet of Things Security." *IEEE Security & Privacy*, *17*(5), pp. 47-57, 2019. DOI: 10.1109/msec.2019.2923973.

[13] P. K. Malik *et al.*, "Industrial Internet of Things in Industrial Revolution 4 . 0 : A State-of-The art in Review," *Computer Communications*, vol. 166, no. March 2021, pp. 125–139, 2019, DOI: 10.1016/j.comcom.2020.11.016.

[14] M. Alsahli, M. Almasri, M. Al-Akhras, A. Al-Issa, and M. Alawairdhi, "Evaluation of Machine Learning Algorithms for Intrusion Detection System in WSN," International Journal of Advanced Computer Science and Applications (IJACSA), 12(5), 2021.

[15] http://dx.doi.org/10.14569/IJACSA.2021.0120574.

[16] G. Fortino, M. Hassan, M. Zhou, A. Goscinski, M. Bhuiyan, J. Li, and S. Bhattacharya. "Guest Editorial Special Issue on Emerging Social Internet of Things: Recent Advances and Applications." *IEEE Internet of Things Journal*, *5*(4), pp. 2478-2482, 2018. DOI: 10.1109/jiot.2018.2860339.

[17] A. Praseed and P. S. Thilagam, "Multiplexed Asymmetric Attacks : Next-Generation DDoS on HTTP / 2 Servers," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1790–1800, 2019, DOI: 10.1109/TIFS.2019.2950121.

[18] F. Alsattam, M. Al-Akhras, M. Almasri, and M. Alawairdhi, "Rule-Based Approach to Detect IoT Malicious Files," Journal of Computer Science, 16(9), 2020.

[19] A. R. Calibration, D. Using, and E. Intelligence, "A Remote Calibration Device Using Edge Intelligence," *Sensors*, vol. 22, no. 1, pp. 1–17, 2022.

[20] F. Seidel and C. Meinel, "Deep En-Route Filtering of Constrained Application Protocol (CoAP) Messages on 6LoWPAN Border Routers," in *IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 201–206.

[21] M. Gidlund, G. Hancke, M. Eldefrawy, and J. Akerberg. "Guest Editorial: Security, Privacy, and Trust for Industrial Internet of Things." *IEEE Transactions On Industrial Informatics*, *16*(1), pp. 625-628, 2020. DOI: 10.1109/tii.2019.2953241.

[22] D. Palma, "Enabling the Maritime Internet of Things : CoAP and 6LoWPAN Performance Over VHF Links," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5205–5212, 2018, DOI: 10.1109/JIOT.2018.2868439.

[23] *MQTT Version 3.1.1 Plus Errata 01*. Edited by Andrew Banks and Rahul Gupta. 10 December 2015. OASIS Standard Incorporating Approved Errata 01. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html.

[24] A. Oram, "Peer-to-peer: Harnessing the Benefits of a Disruptive Technology," O'Reilly, 2001.

[25] P. Millard, *XMPP Protocol XEP-0060: Publish-Subscribe.* XMPP Standards Foundation 2004.

[26] I. Fette, *The WebSocket Protocol.* Hampton, UK: Google, Inc, 2011.

[27] A. Bairagi and D. Chakroborti, "Trust based D2D communications for accessing services in Internet of Things," *2015 18th International Conference on Computer and Information Technology (ICCIT)*, 2015, pp. 50-54, DOI: 10.1109/ICCITechn.2015.7488041.

[28] E. Rescorla, "*Datagram Transport Layer Security Version 1.2.* Palo Alto, CA: RTFM, Inc.

[29] K. Shelby, "The Constrained Application Protocol (CoAP)," CA, USA: Internet Engineering Task Force (IETF), 2014.

[30] R. Thubert, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch," Sophia Antipolis, France: Internet Engineering Task Force (IETF), 2016.

[31] A. Haroon, S. Akram, M. A. Shah, and A. Wahid, "E-Lithe: A Lightweight Secure DTLS for IoT," *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1-5, 2017, DOI: 1.1109/VTCFall.2017.8288362.

[32] A. Haroon, S. Akram, S. Ali., and A. Wahid. "E-Lithe: A Lightweight Secure DTLS for IoT." *IEEE*, *23*(123), pp. 5, 2017.

[33] I. Alnuman, and M. Al-Akhras, "Machine Learning DDoS detection for Generated Internet of Things Dataset (IoT Dat)," 2020 International Conference on Computer and Information Sciences (ICCIS), Jouf University, Jouf, 13-15 October 2020.

[34] F. Montori, L. Bedogni, and L. Bononi. "A Collaborative Internet of Things Architecture for Smart Cities and Environmental

Monitoring." *IEEE Internet Of Things Journal*, *5*(2), pp. 592-605, 2018. DOI: 10.1109/jiot.2017.2720855.

[35] B. Pourghebleh, K. Wakil, and N. Navimipou. "A Comprehensive Study on the Trust Management Techniques in the Internet of Things." *IEEE Internet of Things Journal*, *6*(6), pp. 9326-9337, 2019. DOI: 10.1109/jiot.2019.2933518.

[36] M. Al-Akhras, M. Alawairdhi, A. Alkoudari, and S. Atawneh, "Using Machine Learning to Build a Classification Model for IoT Networks to Detect Attack Signatures," International Journal of Computer Networks & Communications (IJCNC), 12(6), 2020.

[37] L. Chettri, and R. Bera. "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems". *IEEE Internet Of Things Journal*, *7*(1), pp. 16-32, 2020. DOI: 10.1109/jiot.2019.2948888.