Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06823-9

ORIGINAL ARTICLE

A generic and lightweight security mechanism for detecting malicious behavior in the uncertain Internet of Things using fuzzy logic- and fog-based approach

Syed Rameem Zahra[1] · Mohammad Ahsan Chishti[2]

Received: 6 March 2021 / Accepted: 4 December 2021

Abstract
Inspired by the massive surge of interest in the Internet of Things (IoT), this work focuses on the kinetics of its security. By automating everything, starting from baby monitors to life-saving medical devices, IoT brought convenience to people's lives and rapidly became a trillion-dollar industry. However, the future of IoT will be decided on how its security and privacy concerns are dealt with. It is a fact that at present, the security of IoT is lacking in coherent and logical perspectives. For example, the researchers do not adequately accommodate the uncertainty and insider attacks while developing the IoT security procedures, even though most security concerns related to IoT arise from an insider and uncertain habitat. This paper provides a critical analysis of the most recent and relevant state-of-art methods of IoT security and identifies the parameters that are crucial for any security posture in IoT. Considering all the intricate details of IoT environments, this work proposes a Generic and Lightweight Security mechanism for detecting malicious behavior in the uncertain IoT using a Fuzzy Logic- and Fog-based approach (GLSF^{2}IoT). It is developed on the principle of "zero trust," i.e., trust nothing and treat everything as hostile. While Fuzzy Logic has been used to remove uncertainties, the Fog-IoT architecture makes GLSF^{2}IoT inherently better than the cloud-IoT. Once the malicious activity is detected, GLSF^{2}IoT automatically limits the network access against the IoT device that initiated this activity, preventing it from targeting other devices. We evaluated GLSF^{2}IoT for blackhole, selective forward, collusion and DDoS attacks, i.e., attacks which can invalidate any IoT architecture. Besides yielding better accuracy results than the existing benchmarks, we found that GLSF^{2}IoT puts extremely low pressure on the constrained nodes, is scalable, supports heterogeneity, and uncertainty of the IoT environments.

Keywords Fog-IoT · Fuzzy logic · Blackhole attack · Collusion attack · DDoS attack · Insider attacks

1 Introduction

IoT quickly moved from being a buzzword to reality by making people's lives a bit more pleasing and comfortable [1]. However, at present, IoT security, considering the very real and increasing security threats, is woefully inadequate at best and non-existent at worst [2]. Due to their limited storage and processing capacity, and the "walled off" nature of their architecture, most IoT devices cannot run anti-viruses or other security patches, turning them into sitting ducks for cyber-attackers [3]. Recent attacks like Mirai, Wicked, Hajime, Katana, and Amnesia: 33 vouches for the credibility of this statement [4, 5].

In the last phase of 2019, we woke up to headlines like "ring camera hacked by a man in 8-year-old girl's room, taunted her by saying that I am Santa Claus [6]," and "Ring safety camera hacks see home-owners exposed to racial violence and requests for ransom [7]." Researchers have been warning about the vulnerabilities of smart TV's and cameras right from the early days of IoT, i.e., 2013 [8]. Today after 7 years, the attackers and, as such, the attacks are more sophisticated, and yet, the individuals hacking

✉ Syed Rameem Zahra
rameemzahra_36phd17@nitsri.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Srinagar, Srinagar, J&K, India

[2] Department of Information Technology, Central University of Kashmir, Ganderbal, J&K, India

Published online: 23 January 2022

into ring cameras were not highly technical or using Artificial Intelligence (AI). They were people who discovered the credentials on the dark-web by chance and bought them to attack insecure IoT devices [9]. That is, these criminals did not hack-in but simply logged-in using stolen or phished credentials. With this, we can only imagine what a qualified cyber-attacker can do to a "billion-device large" insecure IoT network [10]. As per the verified market research [11, 12], the magnitude of the global IoT market was estimated at USD 212.1 billion in 2019, and is projected to hit USD 1.6 trillion by 2025. The research established the expense of IoT hacks to account for 13.4 percent of annual revenue among businesses with under $5 million in revenue [13]. Users of KrebsOnSecurity lost $323 K when the Distributed Denial of Service attack (DDoS) was launched on the website. In addition to the loss of an 8% customer base of Dyn, the revenue loss caused by the Mirai botnet stands at a staggering $110-million [3].

Many attempts have been in the literature to address the Security and Privacy (S&P) concerns of IoT. A major weakness in most state-of-art security mechanisms is that they put intense pressure on the outsider attacks, i.e., they follow a "castle and moat" mentality [14], often ignoring the terrible insider attacks. If the most recently launched high-profile attacks have taught us anything, it is that insider attacks can render any secure surface-implementing encryption and authentication schemes meaningless [14]. Insider attacks are defined by negligent or corrupt employees and credential thieves [15]. According to the research conducted by Ponemon Institute, a total of 159 organizations from North America (The United States and Canada), Europe, the Middle East, Africa, and Asia–Pacific regions experienced a gigantic 3269 insider attacks over 12 months [15]. These companies focused on securing their parameters while presuming that everything inside is "trustworthy." Figure 1 demonstrates the revenue loss caused by these insider profiles for various activity centers.

These statistics tell us that in order to save the enormous sector of IoT from a total catastrophe, the need of the hour is to quickly deal with its S&P concerns [2, 16]. To acknowledge the concern of insider attacks, a security solution should use the approach of zero-trust, i.e., it must see the insider/authenticated nodes and outsider nodes with equal doubt until they earn that trust.

Besides insider attacks, one of the most neglected among IoT's S&P challenges is the uncertainty of the IoT environment. Humans reason in approximate terms rather than precise [17]. For example, the answers to questions like Ali are slimmer than most of his colleagues. How slim is Ali? 68.8% of India's population live on less than $2 per day [18]. Ambani lives in India. What can be said about Ambani's income? In which manner will a child crawl to

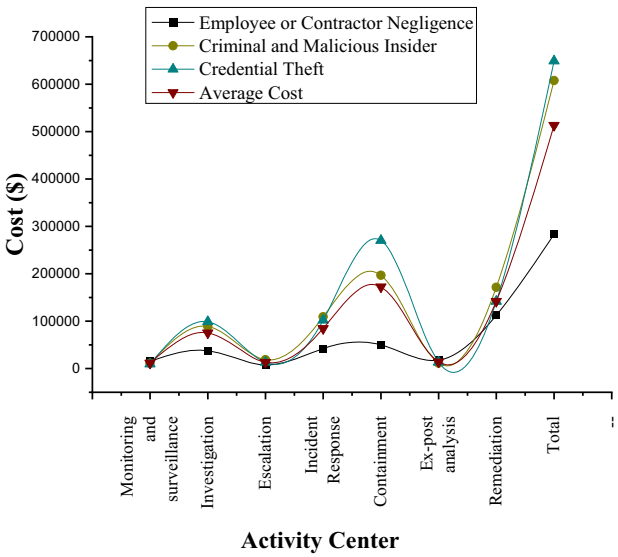


Fig. 1 Revenue loss caused by insider attacks in different activity centers [15]

reach its toy? Shall all be uncertain and approximate. The information is always incomplete and imperfect in an ecosystem created by humans [19]. Hence, it also applies to the world of IoT.

Some illustrations from the world of IoT include; 70% of IoT devices are insecure [20], abc is one such device. How secure is abc? Similarly, their pattern of mobility has uncertainty about their position or membership in any particular group/cluster. A mobile IoT device can belong to multiple groups/clusters simultaneously. Moreover, how IoT consumers express their privacy concerns is also uncertain. For instance, inquire of the owner of a smart camera or a TV how frequently they modify their device's password. A response such as "often" or "rarely" is insufficient, because a normal person's idea of security may be warped, and s/he may lack the qualifications to refer to any number as "frequently" or "rare." That instance, where two password changes each month may appear to be "frequent" to the average user, they may be "rare" in actuality. In general, the IoT world is fraught with such uncertainties. This is because, in most IoT situations, we see a partiality of fact and partiality of possibility. The IoT environment is dynamic, because the information here can sometimes be imperfect, incomplete, or contradictory. Therefore, the answers cannot be precise and drawn by logical systems using precise reasoning [21]. Hence, until we follow our viewpoint on IoT adjustments, an approach where the S&P of IoT is deliberated a critical necessity in the planning stage itself, any attack should only come as a shock and not a surprise.

1.1 Fuzzy logic and fog computing

In this study, we have tried to address the uncertainty of IoT environments by using Fuzzy logic, which is essentially a precise logic of imprecision and approximate reasoning. More precisely, we chose to use fuzzy logic to deal with uncertain IoT, because it is possible to interpret fuzzy logic as an effort to mechanize two extraordinary human capabilities. First, it can converse, think, and make logical choices in an imprecise, imperfect, and ambiguous environment [22]. Secondly, it can execute a massive range of physical and mental jobs without requiring complex estimations, and calculations [23, 24]. For example, in the password dilemma stated above, a fuzzy logic foundation can be used to fuzzify data, determining whether it is frequently or seldom modified depending on a numerical value provided by the owner. In the query of IoT device security, an answer yielded by fuzzy logic might be expressed as: the possibility of abc to be vulnerable to an attack is 0.7. The application can then decide if 0.7 vulnerability is acceptable to it or not. To this end, we propose using Generic and Lightweight Security Mechanism for Detecting Malicious Behavior in the Uncertain IoT using a Fuzzy Logic- and Fog-based Approach (GLSF^{2}IoT). Apart from fuzzy logic, we propose the use of fog nodes in between the edge and cloud nodes. This, in addition, to branching out the workload from the cloud, helps in bringing support to geographical diversities, time-constrained, mobility, and location-aware S&P applications of IoT. All these inherent advantages are missing from the cloud-based IoT architectures [16]. As such, Fog-based IoT provides a distributed, decentralized, and heterogeneous computing environment, pushing the cloud services closer to the network's edge devices. As a result, fog-based IoT brings cloud services closer to network edge devices. Most S&P approaches offered by researchers ignore these considerations.

1.2 Contribution and structure of paper

The distinguishing characteristics of GLSF^{2}IoT that make it different from other IoT security approaches are summarized as follows:

- It tolerates the diversity of IoT devices. Most of the work from the literature on IoT security is inspired by Wireless Sensor Networks, making them inadequate for IoT.
- The security function is pushed away from the constrained edge nodes to let them perform their intended functions and save their crucial resources. It is mainly implemented on cloud and fog layers, thus making it lightweight for edge devices.
- It gives better results for detecting blackhole, selective forwarding, collusion, and DDoS attacks in the IoT environment.
- One of the inherent features of IoT networks is that they are ubiquitous. The overwhelming majority of security procedures, although proposed for IoT, are not scalable to it. GLSF^{2}IoT has been tested on this parameter and has yielded consistent results.
- Most IoT devices use a single-threaded microcontroller with a 2 MB Random Access Memory (RAM) that is insufficient to run a full-fledged operating system or even a simple anti-virus [3] (common-touch requires 128 MB RAM). GLSF^{2}IoT puts negligible memory overhead on them.
- By using fog-based IoT architecture, it can handle the breaches that a mobile node may create.
- Using Fuzzy logic, it is the first of its kind to deal with uncertainties in the IoT atmosphere.
- GLSF^{2}IoT initially assigns zero trust to all the nodes, i.e., insider (authenticated) and outsider nodes, to resist the insider attacks. The monitoring procedures run continuously to identify any misbehavior (an authorized and authenticated node can become corrupt at any time.)

The remainder of the paper is organized as follows: Sect. 2 identifies the motivation for choosing the attacks studied in this paper. To extrapolate the significance of our work, it also presents a critical analysis of the most recent, and relevant state-of-art methods discussing their advantages and shortcomings from an uncertain IoT environment perspective. Section 3 provides a detailed description of our proposed security posture, i.e., Generic and Lightweight Security Mechanism for Detecting Malicious Behavior in the Uncertain IoT using a Fuzzy Logic- and Fog-based Approach (GLSF^{2}IoT). Section 4 analyzes the efficiency of GLSF^{2}IoT under the effect of blackhole, selective forward, collusion and DDoS attacks. It weighs its performance on the parameters of heterogeneity, scalability, architecture employed, energy overhead, memory overhead. In Sect. 5, we have evaluated the suitability of GLSF2IoT to other IoT attacks. Section 6 extracts the conclusion and predicts an immediate future direction for research.

2 Significance of launching the chosen attacks and related work

In this section, we briefly explain the motivation for choosing the blackhole, selective forwarding, collusion and DDoS attacks in accessing the performance of our proposed security posture, i.e., GLSF^{2}IoT. The most recent

and relevant state-of-art solutions proposed for these attacks and the problems linked with those solutions are also discussed.

2.1 Motivation for choosing blackhole and selective forwarding attacks

In IoT application areas like healthcare, intelligent transportation systems, incident response systems, etc., reliable data delivery is important. If this objective is nullified, the whole purpose of having it is lost [25]. Both blackhole and selective forwarding attacks can cause such downfalls to the systems, and as such, they can invalidate any IoT architecture.

In a blackhole attack, for example, an attacker node broadcasts that it has the shortest path to the destination. Lured by this temptation, the source node sends the data. On reception, the attacker node drops all of its data instead of forwarding it to the destination [26]. A blackhole intruder can also imitate the wireless router that links the rest of the network to the IoT devices. Then, with an authorized wireless router, it can redirect the traffic.

In a selective forwarding attack, an attacker node sits somewhere in the selected path. The attack can be launched in two ways, viz. simple and cooperative [27]. A selective forwarding attack can be simple where one malicious node pretends to sit at the median of the shortest path to the destination. It forwards some data packets and drops others just to confuse the source about its credibility. In a cooperative selective forwarding attack, different attacker nodes work in tandem to launch the attack. To gain trust, the first attacker node forwards the data packets, showing a Packet Delivery Ratio (PDR) of 100%, to the other malicious node that drops all of them on reception.

2.2 Motivation for choosing collusion attack

Multiple devices conspire together to bring down the reputation of a particular node. Nodes that conspire to forge a specific object's trust value can do so by falling into one of four potential classifications, viz. Same owner, same geography (all reside in the same place), same workplace (different owners, same workplace), or same social circle (different owners, similar interests, i.e., one community) [28].

It is incredibly challenging to handle collusion attacks in IoT because of the myriad of connected devices in its uncertain and dynamic landscape. The groups keep on changing because of the continuous mobility of the devices [29, 30]. As such, the nodes on which detection procedures run do not get enough time to detect the attack. The use of close and powerful fog nodes in GLSF^{2}IoT help to monitor and detect these attacks as the nodes are scanned every time, no matter where they reside.

2.3 Motivation for choosing DDoS attack

Among the most intriguing cyber-attacks, DDoS attack tries to cause a capacity overload of the server by flooding it with incessant requests, and make it ignore the legitimate customer requests. Several security systems have evaluated their effectiveness on DoS, but it is insufficient and paltry for an IoT network that provides the attackers with a huge attack surface. That is, several devices are likely to be utilized to assault a network, necessitating validation of the framework against a DDoS attack to ensure its relevance.

Moreover, many IoT applications are based on real-time inputs, such as autonomous vehicles, industry, etc. In such situations, the unavailability of a server is catastrophic. For example, when an autonomous vehicle is on the road, and an IoT DDoS attack is launched on the server, it may stop being steered by sensory inputs. The massive IoT network, resource constraints, and diversity of IoT devices make it a duck soup for professional intruders to launch the IoT server DDoS attack [8]. IoT resource constraints make cryptography less suitable due to the time, power, processing cycles, and memory needed to run these algorithms.

2.4 Critical analysis of state of art methods

Over the years, many researchers have attempted to address the S&P issues of IoT networks and devices. Here, we critically examine a considerable number of such security postures to highlight their contributions and clarify how the provided work advanced the state-of-the-art. We also point out their major shortcomings from IoT's S&P perspective.

The major achievement of Seyedi [26], for example, is that they developed a smart agent-based mechanism for blackhole attack that gave less than 19.4% False Positive Rate (FPR), a False Negative Rate (FNR) of less than 22.2% and a Detection Accuracy of 80.5%. [31] Identify blackhole attacks that occur during wireless communication between the base station and nodes with a Detection accuracy of 87.72%. [32] uses genetic programming for detection of Hello flood, Version number, sinkhole, and blackhole attacks. It provides a Detection Accuracy of 92% for the Blackhole Attack with a True Positive Rate (TPR) of 94.7% and FPR of 0.7%. However, it uses the inherent security advantages of Routing Protocol for Low Power and Lossy Networks (RPL) to detect attacks. If a protocol disruption attack happens, the entire scheme gets nullified.

The authors of [27] employed a cryptographic authentication mechanism for identifying the selective forward attack. Their mechanism gave a Detection Accuracy of

94.5%, an FPR of 14.1% and an FNR of 17.5%. To identify the blackhole and selective forward attacks, the authors of [33] developed a lightweight heartbeat protocol. They claim it to be the best approach for detecting these attacks. However, it simples works on ICMPV6 echo requests. If the response isn't received, it implies a blackhole attack. If ICMPV6 is filtered, it identifies selective forward. A very basic application is used, and that too gives a 10% communication overhead. To detect collusion attacks, Yaseen [29] uses SDN to collect data and claims to detect malicious devices during their movement among clusters. It is also scalable to IoT networks.

Svelte, given in [34], is the first Intrusion Detection System (IDS) designed for IoT networks with a TPR of 85% for selective forward and sinkhole attacks. [35] Aims to promote the minimization of FPR for DoS and Botnet attacks, increase the detection rate under distributed attacks, and minimize the workload for the end host. It shows less memory overhead and less power consumption. [36] Detects DDoS and Collusion threats at a faster rate by using the ELM-based Semi-supervised Fuzzy-C Means mechanism (ESFCM). It has a Detection time of 11 ms and an Accuracy of 86.53%. It doesn't take the real-time data traffic, but instead works on the NSL-KDD dataset. ESFCM is just a post-attack observation.

An IDS was designed by Haripriya [37] using fuzzy logic with an FPR of 0.66%. It was implemented for the DDoS attack in IoT setup. However, it gives no clue about how Connection Message Ratio (CMR) and Connection Acknowledgment Message Ratio (CAMR) input parameters were set. Finally, [38] used Fuzzy & Taylor-elephant herd optimization. Three intrusion detection search databases (Db) are used viz. KDD cup, Db-1, and Db-2. The system gives a Detection Accuracy of 93.8% for DDoS attack, but it did not deal with the real attack scenarios.

The critical analysis of these methods is tabulated in Table 1. It identifies the major shortcomings of these procedures.

It is observed that mostly the solutions are developed to identify one type of attack only. This is insufficient and scarce for an IoT network that provides the attackers with a massive extortion landscape. Heed must be paid that any attack is not being launched by the authenticated insiders. It was seen that most articles have missed dealing with this dimension also.

It was also observed that none of the studied approaches consider uncertainty of the IoT landscape while recommending a security mechanism. Also, very few postures ensure that edge devices do not lose all their energy in guaranteeing security, i.e., lightweight. Besides, it is essential to remember that in IoT scenarios, sensors are not viewed as parts of the sensor network connected to the Internet using gateways, as in WSNs. Sensor nodes are instead considered as nodes of the Internet [28, 39, 40]. Therefore, the solutions designed for WSN architectures won't fit the heterogeneous IoT models. The symbols pro $^{(*p)}$ or con $^{(*c)}$ are placed to refer to the existence of the parameter addressed concerning its collective worth. GLSF^{2}IoT is developed to address these issues on the lines of the crucial parameters (listed in Table 1) that are expected of a security procedure meant for IoT.

3 Proposed system

If the recently launched cyber-attacks like SolarWinds [41], Amnesia: 33, etc., have anything to say, it is that there is a continuous rise in the determination and sophistication of attackers. Till the time we establish a solid and reliable security cover to build cyber resilience, there exist illuminated paths to the downfall of IoT. In essence, it becomes imperative to develop a security posture that is built, keeping in view all the essential parameters of IoT security. Built on this ground, Fig. 2 demonstrates the general structure for the proposed GLSF^{2}IoT.

GLSF^{2}IoT architecture is built on 3 layers, viz. edge device layer, fog layer, and cloud layer. The devices in each layer perform specific security tasks in sync to achieve high and real-time detection of attackers. For example, the cloud layer nodes run a hybrid of fuzzy logic-based trust management and anomaly-based detection methods for identifying malicious behavior in the fog layer of the IoT network and to identify the most trusted fog nodes. Also, before operating the security measures, it is critical to group multiple edge nodes with trusted fog nodes to reduce load and boost the system efficiency. For the reasons already described in the study, the grouping has been done to cover the belonging of a device to multiple groups at one time. The trusted fog nodes run unknown attack detection and flag generation procedures. Each layer consults its specific defense library for making various decisions. The following sub-sections go into detail about all the mechanisms that our GLSF^{2}IoT architecture employs.

3.1 Fuzzy logic-based trust management mechanism

Malicious nodes sometimes behave normally to trick the neighbors into believing that they are honest, or to escape the punishment of being thrown out of the network. That is, an attacker displays uncertainty or irregularity in its behavior. To deal with this irregularity, we employ fuzzy logic in our trust management mechanism. This procedure is used to identify the most reliable fog nodes that can build up the local anomaly detection system of GLSF^{2}IoT.

Table 1 Review of the related schemes

Reference	Architecture employed	Insider attacks considered?	Scalable to IoT?	Uncertainty dealt?	Zero-trust policy?	Generic?	Lightweight?
Seyedi [26]	WSN{*c}	No{*c}	No.{*c} IoT architecture not considered. The pressure of security put on IoT nodes	No Routing-protocol-based approach.{*c}	No. Intermediate nodes are also trusted for calculating trust.{*c}	No{*c}	No.{*c} Resource-constrained edge devices perform the calculations
Srinavas [31]	IoT-Cloud{*p}	No{*c}	Yes{*p}	No.{*c} Protocol-based approach	No{*c}	No.{*c} Developed for IoT-based civil construction systems	No.{*c} Pressure of security on constrained edge devices
Qureshi [32]	IoT-Cloud{*p}	No.{*c}	Yes{*p}	No.{*c} It considers the static topology of 40 nodes, which is hardly the case with IoT	No.{*c}	No Performs evaluation on the static sky-motes only.{*c}	Yes.{*p} It cashes the security features of RPL only
Mabodi [27]	Architecture not described{*c}	No.{*c}	No. Architecture not defined.{*c}	No.{*c} Works on black and white principle	No.{*c} Once authenticated, full trust is levied on nodes	No.{*c} Not tested under heterogeneity	No.{*c} Running authentication procedures on resource-constrained edge devices is resource-intensive
Ribera [33]	Single-layer IoT{*c}	No.{*c}	No.{*c} Complexity of IoT architecture not considered	No.{*c} Doesn't even model a real IoT environment	No.{*c} Trusts everything if ICMPV6 response is obtained	No.{*c} Not tested	Yes.{*p} Very basic Doesn't talk about the efficiency of the result
Yaseen [29]	IoT-fog{*p}	No.{*c} It doesn't monitor the fog nodes	Yes{*p}	No{*c}	No.{*c} Complete trust is put on the insider fog nodes	Yes{*p}	Yes{*p}
Raza [34]	IoT-cloud{*p}	No{*c}	Medium.{*p} Developed for static networks Dynamism is the essence of IoT networks	No{*c}	No{*c}	No{*c}	Yes{*p}
Arshad [35]	IoT-Cloud{*p}	No{*c}	Yes{*p}	No{*c}	No{*c}	Yes{*p}	Yes{*p}

Table 1 (continued)

Reference	Architecture employed	Insider attacks considered?	Scalable to IoT?	Uncertainty dealt?	Zero-trust policy?	Generic?	Lightweight?
Rathore [36]	IoT-Fog. {*p} Detection load shared	No.{*c}	Yes{*p}	No.{*c} It is never possible to assess the reliability of a data set when attacks are not launched in real-time Real-time attack has a lot of uncertainty	No.{*c}	NA{*c}	NA{*c}
Haripriya [37]	Single-layer IoT{*p}	No.{*c}	Yes{*p}	No.{*c} Doesn't explain how anomaly detection crisp values were obtained No definition is given for high and low	No{*c}	No{*c}	No{*c}
Velliangiri [38]	Cloud computing environment{*c}	No.{*c}	No.{*c} Computationally extensive Calculations made in 21 hidden layers	No.{*c}	No{*c}	No{*c}	No{*c}

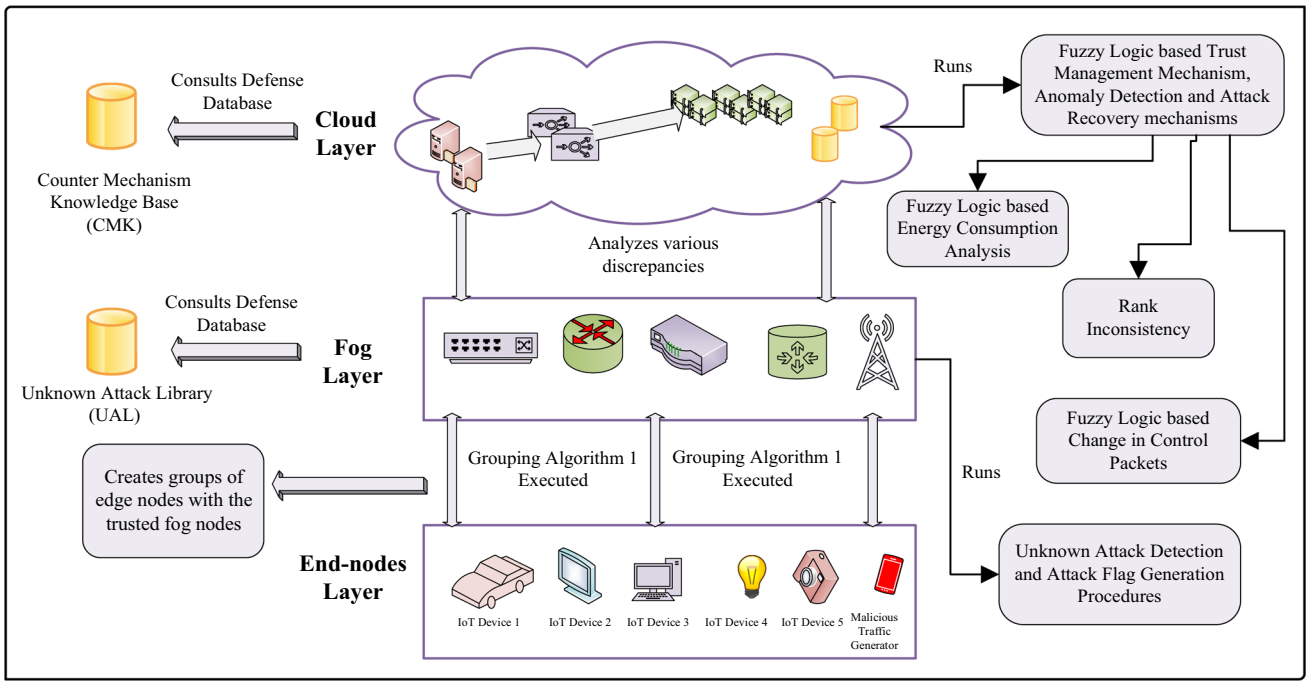


Fig. 2 GLSF^{2}IoT architecture

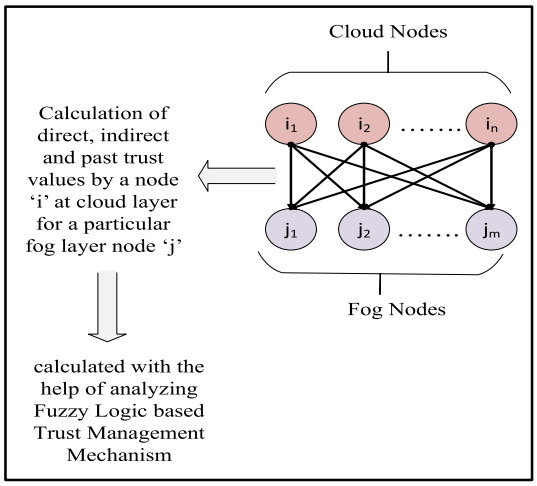


Fig. 3 Fuzzy logic-based trust management mechanism

By creating groups around the reliable fog nodes, the devices will be monitored from the network perimeter itself, and the malicious activities, if detected, will be dealt with quickly. Such a quick remedy is not possible if the cloud is monitoring the entire network [42]. GLSF^{2}IoT can formulate the overall final trust values of different fog nodes by aggregating the trust values received from multiple cloud nodes. It uses a combination of mutual interaction and anomaly-based methods, as shown in Fig. 3.

- *Mutual Interaction Method* It analyzes three parameters for any discrepancies: energy, rank, and control packets. In a network, nodes transmit, forward and receive packets, send routing updates, etc. In GLSF^{2}IoT, the cloud layer nodes monitor these interactions with the fog nodes, detect the inconsistencies (if any) and calculate their trust values. These disparities in interactions are seen by comparing the characteristics of recorded interactions (past) with the present interactions. The past-trust at a time "t," i.e., $T_p(t)$, is calculated using Eq. (1).

$$T_p(t_c) = \frac{\sum_{i=1}^{i=t-1} T_{i,j}^{I}(t)}{t_c - 1} \tag{1}$$

where $T_{i,j}^{I}(t)$ is the trust of node i calculated by node j at present using the method "I," i.e., Mutual Interaction. p refers to the past. The summation is over the trust calculated previously from the beginning to time "t–1," i.e., one less than the current time. For example, if trust is calculated after every one interval and currently the system is in the fourth interval. For the sake of understanding, if we assume that the trust value at $t = 1$ was 0.9, at $t = 2$ was 0.8 and at $t = 3$ was 0.7, then the current past trust value will be equal to $\frac{0.9+0.8+0.7}{3} = 0.7$. By doing this, we are taking all the values into consideration that helps in observing the discrepancies minutely and accurately. As such, the past trust at any moment is calculated by taking the average of all the previously recorded trust values for a particular node. Moreover, as this function is executed by the cloud, computation, and storage of all of these values was not a complex task.

- Also, to calculate the trust values of fog nodes using anomaly detection, the rules given in Table 2 are used. All the reported anomalies are integrated, and depending on the overall sum of detected anomalies, the cloud layer nodes calculate the trust value $T_{i,j}^{A}(t)$ for different fog layer nodes.

Mathematical Formulation The trust value calculated by a node $'i'$ at cloud layer for a particular fog layer node $'j'$ is the weighted sum of trust values calculated with the help of analyzing mutual-interaction with fog nodes $T_{i,j}^{I}(t)$ and the anomalies detected $T_{i,j}^{A}(t)$. Consequently, we get Eq. (2).

$$T_{i,j} = w_1 * T_{i,j}^{I}(t) + w_2 * T_{i,j}^{A}(t) \tag{2}$$

where w_1 and w_2 are the weights associated with respective trust values calculated using mutual-interaction analysis and anomaly detections. These are chosen in a way that the value of $T_{i,j}$ at a particular instant of time "t" always lies in the range $[0, 1]$. Therefore,$T_{i,j} = 1$, indicates complete trust and $T_{i,j} = 0$, depicts complete distrust. The individual trust values in Eq. (2) are calculated as in Eq. (3):

$$T_{i,j}^{Z}(t) = \begin{cases} d \times T_{i,j}^{Z}(t - \delta(t)) + (1-d) \times \left[T_{i,j}^{Z,\text{direct}}(t)\right] & \text{if } i \text{ and } j \text{ are immediate neighbors (direct)} \\ d \times T_{i,j}^{Z}(t - \delta(t)) + (1-d) \times \left[T_{k,j}^{Z}(t)\right] & \text{otherwise (indirect)} \end{cases} \tag{3}$$

$T_{i,j}^{Z,\text{direct}}(t)$ is the trust value calculated by node $'i'$ with the help of method $'Z'$ using direct communication with the node $'j'$, $'Z'$ can take values $'I'$ or $'A'$ depending upon the method used for trust calculation. $T_{k,j}^{Z}(t)$ is the trust calculated for node $'j'$ due to $'Z'$ by some other node $'k'$ in the

Table 2 Anomaly detection rules for trust management module

Rules	Description
Multiple connection rule	An Anomaly is detected if the node sends multiple connection establishment requests
Recursive packet flow rule	An Anomaly is detected if small-sized messages are received repeatedly from a particular fog node
Collision rule	An Anomaly is detected if the majority of packets in the IoT network pass or get forwarded through a particular fog node

network. In other words, it is a combination of direct, indirect, and past trust values. $\delta(t)$ is the update interval for trust and $'d'$ is the system parameter whose value lies in the range $[0, 1]$. Lesser value of $'d'$ indicates more dependence on direct and current values for trust calculation. $T^{I}_{i,j}$ is calculated using discrepancies in interactions.

- *Fuzzification process* For mapping the crisp input variables (direct trust, indirect trust and past-trust) into fuzzy sets, we define triangular and trapezoidal membership functions for trust as (Eqs. 4–6);

$$\mu_H(x;a,b) = \begin{cases} 0; & x<a \\ \dfrac{x-a}{b-a}; & a\le x\le b \\ 1; & x>b \end{cases} \tag{4}$$

$$\mu_A(x;c,d,e) = \begin{cases} 0; & x<c \\ \dfrac{x-c}{d-c}; & c\le x\le d \\ \dfrac{e-x}{e-d}; & d<x<e \\ 0; & x\ge e \end{cases} \tag{5}$$

$$\mu_L(x;f,g) = \begin{cases} 0; & x>f \\ \dfrac{f-x}{f-g}; & g\le x\le f\, x<g \\ 1 \end{cases} \tag{6}$$

here $\mu_R(x)$ where $R = \{H, A, L\}$ is not a probability value, but a subjective judgment of grade-a membership. It says to what degree the element x in R belongs to a fuzzy set (H, A or$L$). These functions are chosen, because they are proved to be computationally less intensive for sensor nodes.

The values for direct, indirect, and past trusts are calculated using the region boundaries (Table 3), and as such,

Table 3 Region Boundaries for membership functions

Trust inputs	a	b	c	d	e	f	g
$T^{z}_{i,j}$	0.6	0.8	0.1	0.5	0.8	0.5	0.1
$T^{z}_{k,j}$	0.5	0.8	0.1	0.4	0.8	0.5	0.1
T_p	0.6	0.8	0.1	0.5	0.8	0.5	0.1

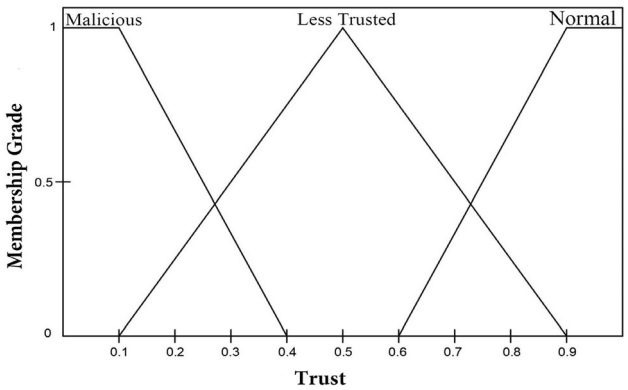


Fig. 4 Membership function for trust

are classified into three degrees, viz. high, average, and low.

- *Development of Fuzzy rule base* The membership functions for demonstrating the trust values and the legitimacy of the obtained overall trust for a fog node are represented by three linguistic labels: Normal, less trusted, and malicious, as shown in Fig. 4. Consequently, the fuzzy rule base is developed as follows:

  If direct trust is low, indirect is low, and the past is low, then the overall trust of a fog node is malicious.
  If direct is high but indirect and past trusts are low, then the overall trust is malicious.

Likewise, 27 different trust values can be obtained for fog nodes. Among these, we choose the fog nodes whose trust values lie in the range of 0.8–1 (highly trusted).

- *Defuzzification* The mean of centroids of gravity gives the crisp value of overall trust for every membership function (Eq. 7)

$$\text{overall trust} = \frac{\sum_{x=a}^{b} \mu_R(x) \times x}{\sum_{x=a}^{b} \mu_R(x)} \tag{7}$$

3.2 Grouping technique employed in GLSF^{2}IoT

Once the trusted fog nodes are identified, our grouping mechanism creates groups of edge nodes with the trusted fog nodes. Algorithm 1 depicts how grouping has been done. The grouping algorithm breaks the uncertainty of one node (edge or fog node) belonging to one crisp group at a time. Because of its mobile nature, a node can belong to multiple groups also. In that case, it will just be monitored by multiple trusted fog nodes.

Algorithm 1 Grouping algorithm running at trusted fog nodes

Require: A list $'C_i'$
if $((Node_{id}\ not\ in\ C)$ and $(pkt_h \leq h))$, then
    Add $Node_{id}$ in $'C'$
    Inform $Node_{id}$
end if
if $(time \geq t)$
    Discard newly received acknowledgement-packet
    Save $'C'$ at the cloud layer
end if

Symbols used:
t – It is the predefined threshold time needed to send a special packet and receive an acknowledgment-packet and is set by a trusted fog node.
h – Threshold Hop-count. Also, set by a trusted fog node.
C_i – List of nodes in the group headed by i^{th} trusted fog node.
pkt_h – Hop-count in received acknowledgment-packet.
$Node_{id}$ – ID of an edge device that has sent the acknowledgment-packet.

3.3 Fuzzy logic-based energy consumption analysis

This is a known fact that IoT edge devices are constrained by energy. If the pressure of security is not taken away from the edge layer, then the purpose for which these devices are actually created is defeated, because their entire energy is lost in doing the security calculations. As such, if a security posture is proposed for an IoT network, one of its prime objectives should be to detect attacks in the network without increasing the burden on edge devices. While designing GLSF^{2}IoT, we have acknowledged this requirement and made sure that it puts negligible energy overhead on the edge devices.

To extrapolate the procedure of fuzzy logic combination in this mechanism, we consider two metrics, viz. Total energy spent by nodes for exchanging information (E_T) and a total number of transmissions ($count$) required to successfully transmit a packet.

Let the total energy consumed by source and destination nodes be represented by E_T. It is given by Eq. (8).

$$E_T = (1 - P_r) \times E_S + E_D \tag{8}$$

where E_S is the energy drained by the sender node, and E_D is the energy exhausted by the destination node. P_r is the probability that the data packets transmitted to a particular destination fail to reach it. $(1 - P_r)$ is the probability of successful transmission. The energy consumed by sender node E_S is calculated using Eq. (9).

$$\begin{aligned} E_S &= T_{cont} \times E_{sensing} + T_{TA} \times E_{sensing} + T_D \times E_{TX} \\ &\quad + T_{TA} \times E_{sensing} + T_{ack} \times E_{RX} \\ i.e., E_S &= E_{sensing} \times (T_{cont} + 2T_{TA}) \\ &\quad + T_D \times E_{TX} + T_{ack} \times E_{RX} \end{aligned} \tag{9}$$

where T_{cont} is the time spent in sensing the carrier, $E_{sensing}$ is the energy spent by a node to sense the channel, T_{TA} is the Turn-around-Time, i.e., a fixed duration for which the device waits on sensing a free channel, T_D is the time spent by device for transmitting data, E_{TX} is the energy spent by a node to transmit the data packet, T_{ack} is the time spent by the receiver to send the acknowledgment, and E_{RX} is the energy spent by the node for receiving a packet. Figure 5 provides a visual description of these times.

Also, the energy spent by a destination node is calculated as (Eq. 10):

$$E_D = T_D \times E_{RX} + T_{TA} \times E_{sensing} + T_{ack} \times E_{TX} \tag{10}$$

where T_D is calculated using Eq. (11).

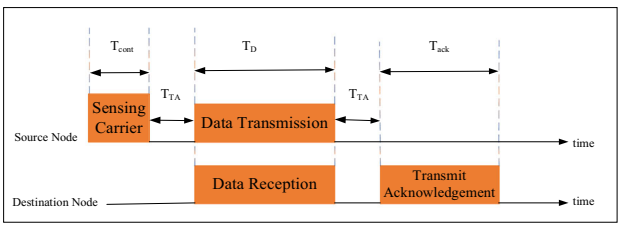


Fig. 5 Times spent by source and destination nodes for various processes

$$T_{\mathrm{D}} = \frac{\text{no of slots} \times \text{no of bits per slot}}{\text{packet transmission rate}} \tag{11}$$

The *count* is defined by three linguistic variables viz. Small (transmission successful in the first attempt), medium, and large (transmission successful after maximum transmissions). The variation in total number of transmissions is categorized into three sets, as shown in Fig. 6. Likewise, we have classified E_T into three sets shown in Fig. 7

- *Fuzzification* Table 4 demonstrates the development of fuzzy rules for identifying the routes with the most reliable nodes. It indicates lesser the *count*, the lesser will be the E_{T}, and hence better will be the route.

Where Good, Very Good, Average, Poor, and Very Poor are the output link qualities. Link Quality refers to the amount of energy consumed by it. *Count*, and E_{T} membership functions can be used to detect the link quality. Mamdani model helps in calculating the medium link quality [43] as follows (Eq. 12);

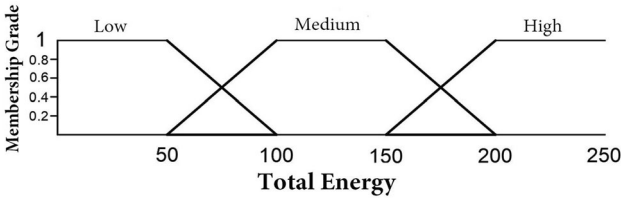


Fig. 7 Membership function for total energy

Table 4 Development of fuzzy rules and output fuzzy matrix

Count/E_{T}	Low	Medium	High
Small	Very Good	Good	Average
Medium	Good	Average	Poor
Large	Average	Poor	Very Poor

$$\text{median(Qual)} = \text{maximum} \begin{cases} \text{minimum(Small(count), High } (E_{\mathrm{T}})) \\ \text{minimum(Median(count), Median}(E_{\mathrm{T}})) \\ \text{minimum (Large(count), Small}(E_{\mathrm{T}})) \end{cases}, \tag{12}$$

For composing, it uses minimum operator, and for aggregation, it uses maximum operator.

- *Defuzzification* The crisp output value is obtained by calculating the mean of centroids of gravity as in Eq. (7). Figure 8 shows the defuzzified output.

After analyzing the packets transmitted in 6LoWPAN networks, we have deduced that the usual period for a single symbol is 16 µsec, and a single slot consists of approximately 20 symbols. To sense a carrier, a 12 symbol slot is used, each comprising of 4 bits. Moreover, we obtained T_{TA} = 192 µsec and T_{count} = 128 µsec

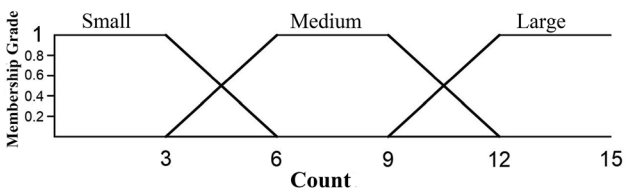


Fig. 6 Membership function for count

3.4 Rank-inconsistencies

RPL is a de-facto IPv6 routing protocol for low power and lossy networks used to find the optimal routing path between source and destination nodes in an IoT network. It constructs a Destination-Oriented Directed Acyclic Graph (DODAG) that depicts the graphical position of all nodes in the network. The Rank of a node defines its position in a DODAG with respect to the root node and relative to other nodes in the network. Some attacker nodes can produce

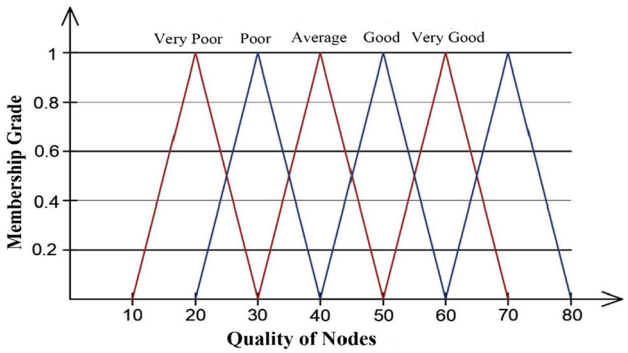


Fig. 8 Defuzzified output for energy variation of a node

wrong information regarding the ranks of different IoT nodes connected to each other, thereby causing inconsistency in the network. These attacker nodes can also propagate wrong information regarding their ranks to attract data traffic towards themselves, and then cause harm to received data. We have proposed algorithms 2 and 3 for detecting rank discrepancies.

Algorithm 2 Rank inconsistency correction during the initial phase of the IoT network establishment

Requires: L_i - The list of nodes in the group with trusted fog node $'i'$
for-each node $'N'$ in L_i, do
  for-each neighbor $'n'$ of $'N'$, do
    $D = N_{rank} - n_{rank}$
    $avg = \frac{D}{2}$
    $if(ang > ang \times 0.2)$, then
      $n_{fault} = n_{fault} + 1$
    end if
  end for
end for
for-each node $'n'$, of $'N'$ do
  $if\ n_{fault} > fault\_threshold$, then
    use another node in the vicinity to calculate the rank of $'n'$
    reset the rank of $'n'$
  end if
end for

Symbols used:
N_{rank} – Rank of node $'n'$ as calculated/predicted by node $'N.'$
n_{rank} – Rank of node $'n'$ as calculated/predicted by the node itself.
n_{fault}– Variable counting the number of inconsistencies in rank for the node $'n.'$
$fault\_threshold$ – Predefined Threshold values for the number of faults.

3.5 Fuzzy logic-based change in control packet anomaly detection

We assume that the IoT nodes use TCP to provide transport layer services and all the data packets pass through a trusted fog node. Since the edge devices are deployed to provide a particular service, therefore only the traffic with specific characteristics should be being transmitted between the edge node and the trusted fog node. The malicious node, however, has to send more packets (to communicate with its command and control center) and packets lengths larger than usual (to launch an attack). Using the characteristics of normal IoT network traffic, we calculated the average length of packets transmitted, and used it as a threshold limit to detect an anomaly. Any deviation in the number or length of packets from a threshold detects a malicious node. The fuzzy rule base for this mechanism is given in Table 5. Input variables form the precedent part of the rule, and output forms the subsequent part. Figure 9 shows the mechanism.

3.6 Cloud's attack recovery mechanism

Using these mechanisms, the cloud identifies the most trusted and malicious fog nodes. Once it detects a malicious fog node, it puts its id in the blacklist and warns the network about it. It is to be noted that these procedures run continuously. That is, a fog node can be the most trusted at a particular time, but it can become corrupt later to launch an insider attack. GLSF^{2}IoT doesn't trust any node forever. If a previously trusted node is observed to behave maliciously, cloud directs the nodes falling in its group to break their connection with it, group with some other trusted fog node, or communicate directly with the cloud. It is associated with a database called Counter-Mechanism Knowledgebase (CMK). It consults CMK for making decisions as given in Eq. (13).

$$\text{CMK} = \{B_{id}, M_{id}, M_{name}, M_{f-base}, A_{id}, lay, ser\} \tag{13}$$

where B_{id} is the list of blacklisted node id's, M_{id}, M_{name}, and M_{f-base} are the id, name, and fuzzy-logic rule base of the mutual-interaction method. A_{id} is the attack id already known to the cloud. *lay and*, *ser* refer to the layer and service that are responded to by the cloud. The dynamic nature of CMK is ensured by continuous updates concerning new attacks identified by fog layer. The CMK update is described in the following sub-section.

3.7 Anomaly-based detection

The responsibility of detecting blackhole, selective forward, collusion, and DDoS attacks was levied on the

Table 5 Fuzzy rule base for detecting anomalies in control packets

Precedent part of the rule		Subsequent part of the rule
Number of packets (per second)	Length of packets (bytes)	Possibility of attack
Large (> 12)	Large (> 240)	High (73–100)
Large (> 12)	Average (220 to 245)	High (73–100)
Large (> 12)	Low (< 223)	Medium (45–83)
Medium (8–13)	Low (< 223)	Medium (45–83)
Medium (8–13)	Average (220–245)	High (73–100)
Medium (8–13)	Large (> 240)	High (73–100)
Small (< 10)	Low (< 223)	Low (23–50)
Small (< 10)	Average (220–245)	Medium (45–83)
Small (< 10)	Large (> 240)	Medium (45–83)

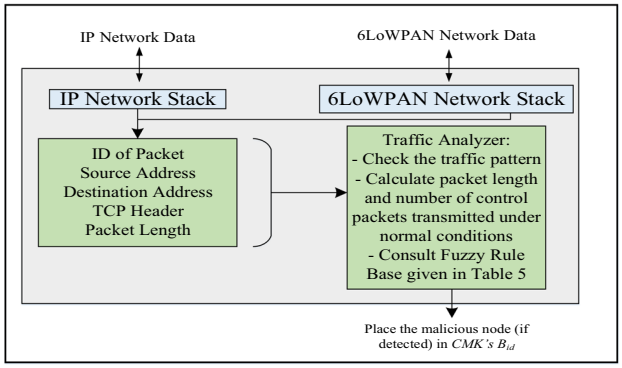


Fig. 9 Fuzzy logic-based change in control packet anomaly detection

trusted fog nodes. To achieve this, we have added two sub-segments to the fog layer, viz. Unknown Attack Detection segment (UAD) and Attack Flag Generation segment (AFG).

- *Unknown Attack Detection segment (UAD)* This segment identifies the threats by consulting the fuzzy logic-based Unknown Attack Library (UAL) given in Eq. (14).

$$\text{UAL} = \{A_{id}, name, def, lay, ser\} \tag{14}$$

where A_{id} is the attack ID,*name* denotes the name of the attack, *def* gives its description,*lay*, *ser* identifies the layer and service type set off by the attack. UAD constantly screens network's working through Forward Packet Ratio (FPR), Average Destination Sequence Number (ADSN), Average Packet Drop Rate (APDR), Internal Resemblance (IR), External Resemblance (ER), and Signal-to-Noise Ratio (SNR) to calculate the Degree of Attack (DoA). This choice of parameters was made, because the literature points out that they can identify any variation of the respective attacks for which they are chosen [40, 44, 45]. Table 6 shows how the parameters have been calculated for the nodes.

Research indicates that every new attack that is tossed is just a 1–2% variation of the existing ones, i.e., zero-day attacks are just the mutated versions of classical attacks

Table 6 Performance parameters for GLSF^{2}IoT

Parameter	Calculation
FPR	No. of packets forwarded/No. of packets received
ADSN	$\text{Destination}_{\text{seq-no}}(t_i) - \text{Destination}_{\text{seq-no}}(t_{i-1})$/no. of destination sequence numbers
APDR	No. of packets dropped/No. of packets received
IR	$\lvert\text{median}\{r_c^w[n_1], r_c^w[n_2], r_c^w[n_3]. \ldots \ldots \ldots, r_c^w[n_m] - R_c(w)\rvert$ $R_c(w)$ is the recommended trust for node $'w'$ in community $'C'$ $r_c^w[n_1]$ denotes the recommended trust of node n_1 for $'w'$ in community $'C'$ Median—The median of trust recommendations for $'w'$
ER	$\lvert\{\text{sum}/\text{count}\} - R_c(w)\rvert$ $\text{sum} = \text{sum} + \text{median}\{R_c(w)\}$ $count = count + 1$ Initially, sum and count = 0
SNR	Signal power/Noise Power

Table 7 Development of fuzzy rule base for blackhole, selective forward, collusion, and DDoS attacks

Black hole attack		
Precedent part of the rule		Subsequent part of the rule
FPR	ADSN	DoA
Low (< 17)	Low (0–5)	High (73–100)
	Medium (2.5–7.5)	High (73–100)
	High (5–10)	Medium (45–83)
Medium (12–32)	Low(0–5)	Medium (45–83)
	Medium (2.5–7.5)	Low (23–50)
	High (5–10)	Low (23–50)
High (> 23)	Low (0–5)	Medium (45–83)
	Medium (2.5–7.5)	Low (23–50)
	High (5–10)	Low (23–50)
Selective forward attack		
Precedent part of the rule		Subsequent part of the rule
APDR	FPR	DoA
Low (< 17)	Low (< 17)	Medium (45–83)
Medium (12–32)		High (73–100)
High (> 23)		High (73–100)
Low (< 17)	Medium (12–32)	Medium (45–83)
Medium (12–32)		High (73–100)
High (> 23)		High (73–100)
Low (< 17)	High (> 23)	Low (23–50)
Medium (12–32)		Medium (45–83)
High (> 23)		Medium (45–83)
Collusion attack		
Precedent part of the rule		Subsequent part of the rule
IR	ER	DoA
Low (0–0.5)	Low (0–0.5)	Low (0–0.5)
	Medium (0.2–0.7)	High (0.5–1)
	High (0.5–1)	High (0.5–1)
Medium (0.2–0.7)	Low (0–0.5)	Medium (0.2–0.7)
	Medium (0.2–0.7)	High (0.5–1)
	High (0.5–1)	High (0.5–1)
High (0.5–1)	Low (0–0.5)	High (0.5–1)
	Medium (0.2–0.7)	High (0.5–1)
	High (0.5–1)	High (0.5–1)
DDoS attack		
Precedent part of the rule		Subsequent part of the rule
SNR	FPR	DoA
High (> 15)	Low (< 17)	High (73–100)
High (> 15)	Medium (12–32)	High (73–100)
High (> 15)	High (> 23)	Medium (45–83)

Table 7 (continued)

DDoS attack		
Precedent part of the rule		Subsequent part of the rule
SNR	FPR	DoA
Medium (1–18)	Low (< 17)	High (73–100)
Medium (1–18)	Medium (12–32)	Medium (45–83)
Medium (1–18)	High (> 23)	Low (23–50)
Low (− 1 to 2.5)	Low (< 17)	Medium (45–83)
Low (− 1 to 2.5)	Medium (12–32)	Low (23–50)
Low (− 1 to 2.5)	High (> 23)	Low (23–50)

[46]. Therefore, by using the knowledge of experts, the fuzzy rule base for detecting these attacks is developed (table 7) for threat detection in IoT. Using a conjunction (AND) operator, these fuzzy rules are executed partly and in parallel. However, if the execution of more than one rule results in the same outcome, the operator of a disjunction is used. Consulting UAL, UAD can classify DoA into 3 degrees viz. Low, medium, and high.

- *Attack Flag Generation segment (AFG)* When the trusted fog node detects a threat, it sends an instant update to the cloud using a Flag function, defined as (Eq. 15):

$$Flag(t) = \langle Attack, Attacker, freq \rangle \tag{15}$$

where Attack refers to the type of attack that is launched, Attacker defines the malicious node (its id and the id of trusted fog node in whose group it currently falls), and freq is the number of times the attack was initiated by this node.

The cloud, on receiving the $Flag(t)$, updates its CMK and propagates this information among all the other trusted fog nodes, and orders them to halt communication with these malicious nodes. The CMK update occurs as per Eq. (16).

$$\mathrm{CMK}(t) = \begin{cases} (c_0, c_1, c_2, \ldots\ldots\ldots, c_n); & t = 0 \\ \mathrm{CMK}(t-1) \cup \mathrm{CMK}_{\mathrm{new}}(t); & t > 0 \end{cases} \tag{16}$$

where $(c_0, c_1, c_2, \ldots\ldots\ldots, c_n)$ is the cloud's primary counter-mechanism knowledgebase, and $\mathrm{CMK}_{\mathrm{new}}(t)$ is the updated one.

4 Simulation results

The GLSF2IoT was instantiated on the open-source Contiki hybrid operating system optimized for IoT, using a workstation equipped with 64 GB RAM and an Intel Xeon processor running at 3.60 GHz. Evaluation was conducted

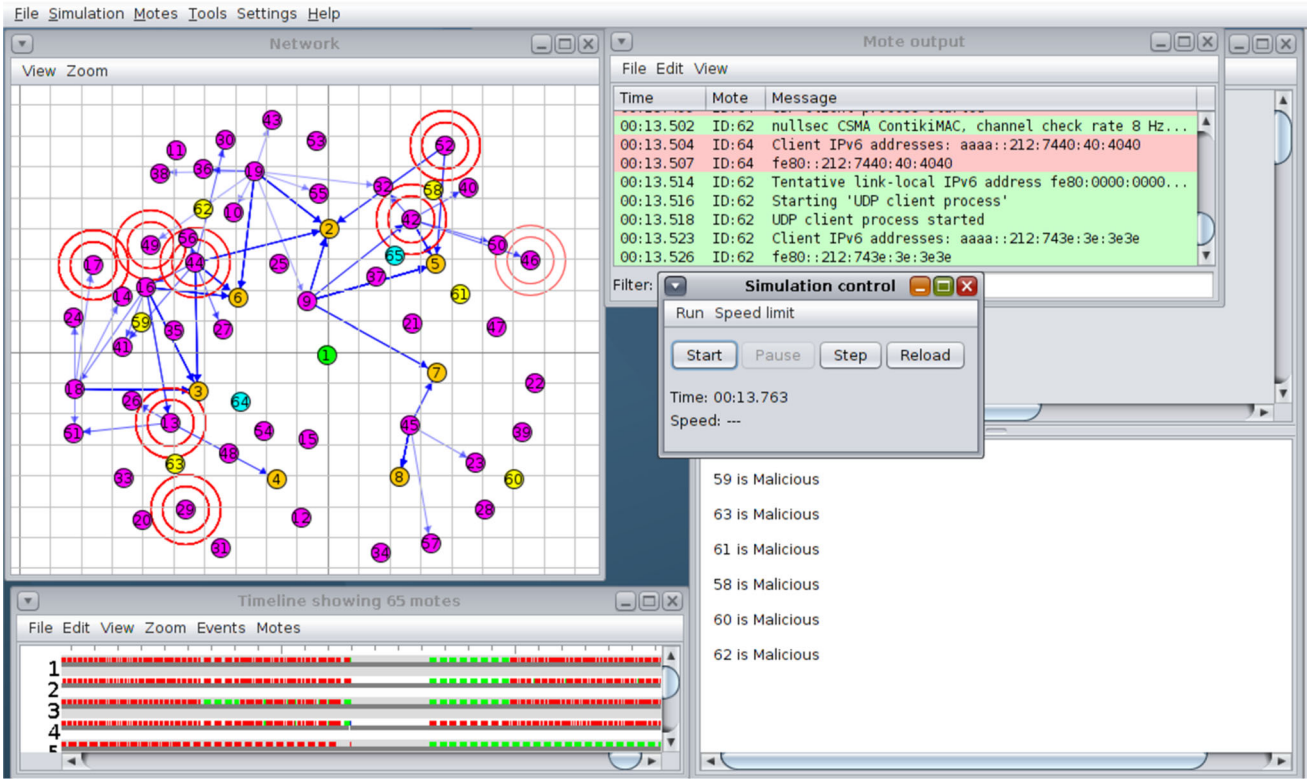


Fig. 10 Experimental set-up

using Cooja, Contiki's simulation tool. Cooja is a flexible simulator conceived to simulate sensor and IoT networks running the Contiki operating system. The properties of Tmote-sky IoT devices were exploited to initiate the attacks. A lossy environment is considered because; the 6LoWPAN protocol used in IoT networks is inherently lossy [16, 34].

Figure 10 shows one of the experimental setups. The Cooja network simulator contains five windows that are studied for calculating the performance metrics of a system. The Network window shows the physical organization of the motes. To construct a topology, the physical positions of the motes could be changed. In the network window, each type of mote has a different color based on its purpose. For example, in Fig. 10, i.e., in the implementation of GLSF^{2}IoT, the cloud node is green, the fog nodes are blue, the edge motes are pink, and the malicious nodes are yellow. Mote properties, radio environment of each mote, mote kind, and radio communication between the motes could all be seen graphically in the network window. The simulation control window allows us to modify the simulation speed as well as pause, restart, and reload the currently running simulation. The note window is used to write the theories and key points of the simulation and store them. The Cooja network simulator also displays a timeline for each sky mote (in our case, 65 motes). IoT networks' power consumption and network traffic could be seen with the timeline. The mote output also displays the similar information but in a different format. It tells what every mote is doing at a particular instant of time with a message. Transmitting radio signals are displayed in blue, receiving radio signals in green, and radio interference in red.

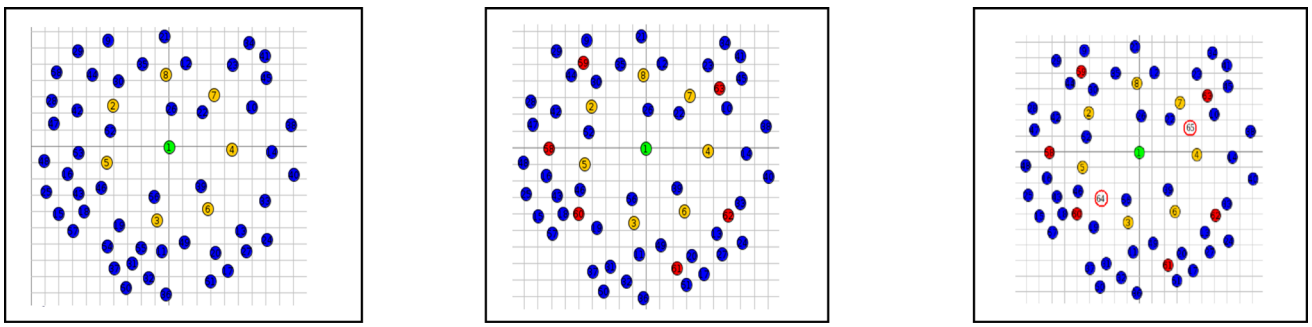
The key parameters concerning the experiments are given in Table 8.

4.1 Simulation environment

Figure 10 shows that GLSF^{2}IoT can detect the attackers in real-time. The edge nodes are normal sky motes with a modified client.c file. The modification was made to allow these devices to send and receive data like normal IoT devices. The sky motes available in Cooja otherwise work only on the control packets. The IoT network with no malicious nodes is shown in Fig. 11 scenario-1. The fog nodes have been created by modifying the udp-server.c file. The cloud node is running the udp-server.c file. In scenario-2 of Fig. 11, we have launched the attacks (by modifying the edge node client files) at the edge layer only. In Scenario-3, a more aggressive attack landscape is created by having attackers at both the edge and fog layers.

Table 8 Parameters for experimental evaluation

Parameter	Value
Access level of the attacker	Active
Attackers approach	1. Target extensive attack landscape 2. Compromise software of vulnerable devices 3. Attack MAC and Network layers
Network area	500 ×500 m^2
Maximum simulation time	50 min
Maximum number of edge IoT nodes	200
Maximum number of fog IoT nodes	40
Maximum number of cloud nodes	1
Number of attack scenarios	3 (60 edge, 10 fog, 1 server, and 20 attacker nodes; 120 edge, 25 fog, 1 server, and 40 attacker nodes; 200 edge, 40 fog, 1 server, and 85 attacker nodes.)
Maximum number of iterations for checking the result accuracies	10
No. of trust recommendations sent (for collusion attack)	100
Max. no of false recommendations sent	50
Node type emulated	Tmote-sky
Radio Medium used	Unit Disk Graph Medium (UDGM) Distance loss
Network environment	Lossy
Ranges of nodes	Rx and Tx: 50 m, Interference: 100 m
PYH and MAC Layer	IEEE 802.15.4
Duty cycle	ContikiMAC
Transport layer	UDP
Network layer	uIPv6, 6LoWPAN
Objective function	ETX



(Scenario-1) **(Scenario-2)** **(Scenario-3)**

Fig. 11 Scenario-1: no malicious nodes (Green—Server node, Yellow—Fog node, Blue—Edge node), Scenario-2: malicious nodes at edge layer only (Green—Server node, Yellow—Non-malicious Fog node, Blue—Non-malicious Edge node, Red—Malicious Edge node) and Scenario-3: malicious nodes at fog layer as well as at edge layer (Green—Server node, Yellow—Non-malicious Fog node, Blue—Non-malicious Edge node, Encircled-Red—Malicious Fog node, Red—Malicious Edge node) (color figure online)

4.2 Performance of comparative analysis of GLSF^{2}IoT under various attacks

The parameters chosen for analyzing the performance of GLSF^{2}IoT below the impact of blackhole, selective forward, collusion, and DDoS attacks are given in Table 9, and the performance of GLSF^{2}IoT is tabulated in Table 10.

The detailed TPR analysis of GLSF^{2}IoT is given in Figs. 12, 13, 14 and 15.

Table 9 Parameters for performance analysis

Parameter	Definition	Calculation
True positive rate (TPR)	No. of malicious nodes that GLSF^{2}IoT correctly detects as malicious	$\frac{TP}{TP+FN} \times 100$
True negative rate (TNR)	No. of non-malicious nodes that GLSF^{2}IoT correctly detects as normal	$\frac{TN}{TN+FP} \times 100$
False positive rate (FPR)	No. of non-attacker nodes falsely detected as attackers	$\frac{FP}{FP+TN} \times 100$
False negative rate (FNR)	No. of attacker nodes falsely detected as non-attackers	$\frac{FN}{FN+TP} \times 100$
Detection accuracy	Percentage of attacks detected by the system	$\frac{TP+TN}{TP+TN+FP+FN} \times 100$

Table 10 Performance of GLSF^{2}IoT under various attacks and uncertain attack scenarios

Attack launched	Performance of GLSF^{2}IoT in Scenario-2	Performance of GLSF^{2}IoT in Scenario-3
Blackhole	Max TPR = 98%	Max TPR = 96%
	Max FPR = 1.4%	Max FPR = 1.6%
	Max FNR = 0.5%	Max FNR = 0.62%
	Max Detection Accuracy = 96%	Max Detection Accuracy = 96.5%
Selective Forward	Max TPR = 96%	Max TPR = 94%
	Max FPR = 1.8%	Max FPR = 1.9%
	Max FNR = 0.8%	Max FNR = 1%
	Max Detection Accuracy = 96.5%	Max Detection Accuracy = 94.5%
Collusion	Max TPR = 92%	Max TPR = 91%
	Max FPR = 0.9%	Max FPR = 0.95%
	Max FNR = 0.85%	Max FNR = 1.5%
	Max Detection Accuracy = 92.5%	Max Detection Accuracy = 92%
DDoS	Max TPR = 99%	Max TPR = 97.5%
	Max FPR = 2%	Max FPR = 2.5%
	Max FNR = 0.75%	Max FNR = 0.79%
	Max Detection Accuracy = 99%	Max Detection Accuracy = 98%

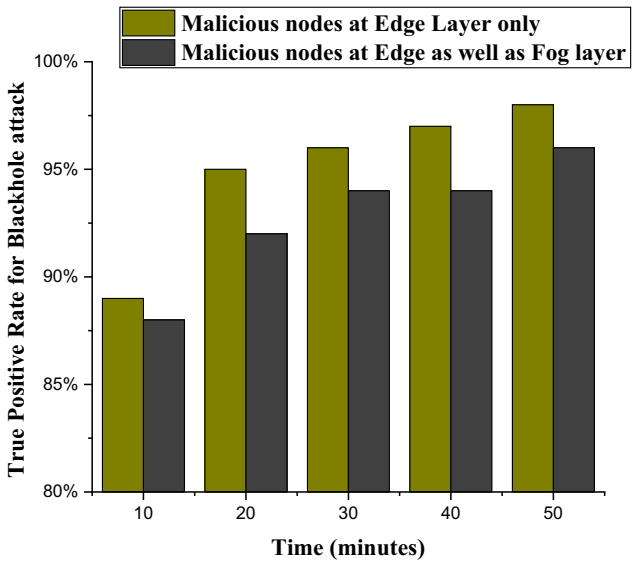


Fig. 12 TPR for blackhole attack in scenario's 2 and 3

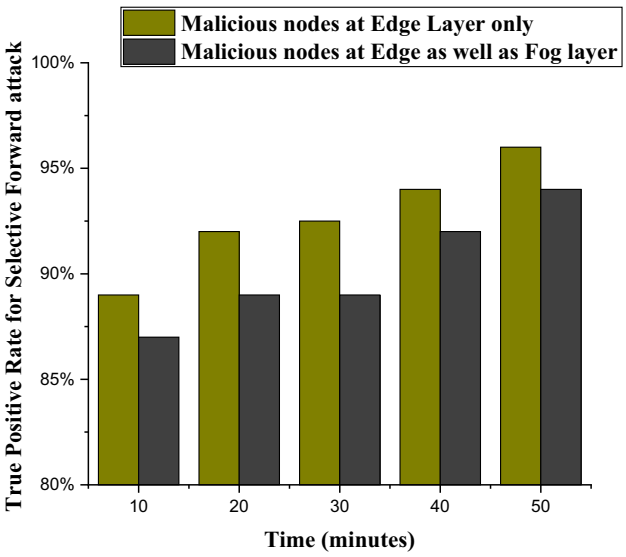


Fig. 13 TPR for selective forward attack in scenario's 2 and 3

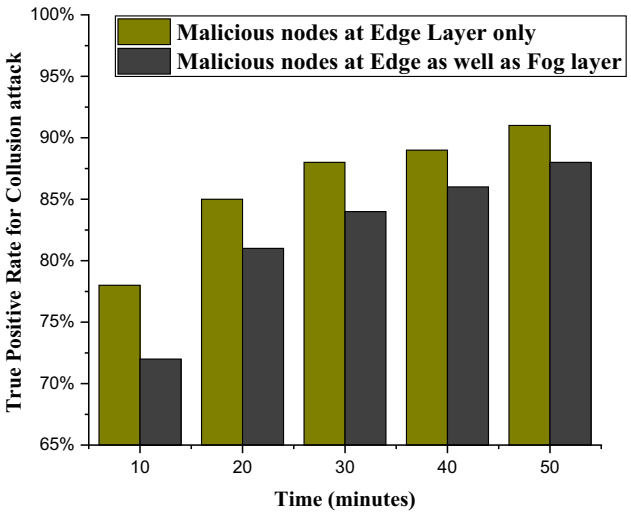


Fig. 14 TPR for collusion attack in scenario's 2 and 3

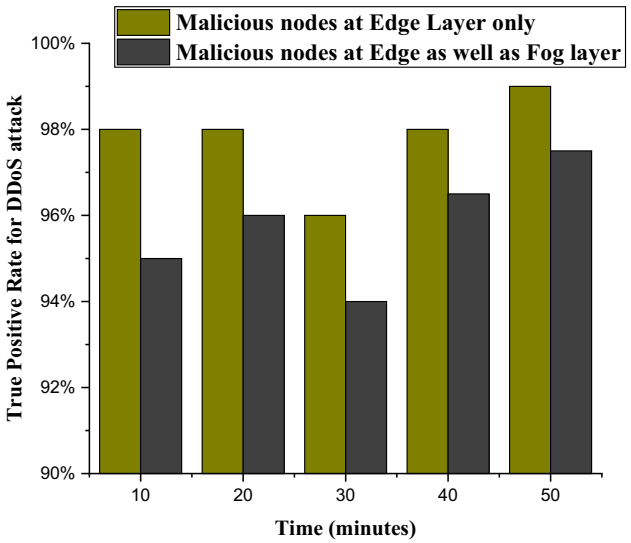


Fig. 15 TPR for DDoS attack in scenario's 2 and 3

It is seen from the graphs that no output is obtained until five minutes. It must be realized that this is the set-up time of the network, i.e., the time required for all units to synchronize. As soon as that is achieved, GLSF^{2}IoT detects the attacks in real-time. Moreover, the fluctuations in the results are due to uncertainty in the time intervals when the attacks get launched. The uncertainty is introduced to make the simulated network closer to the realistic/physical IoT network scenario.

Also, Fig. 16 orchestrates the detection accuracy of individual attacks and the cumulative accuracy of GLSF^2-IoT under the influence of all the attacks launched together.

The results obtained indicate that GLSF^{2}IoT's success in detecting these attacks is impressive and admirable, because even in the highly aggressive attack environment,

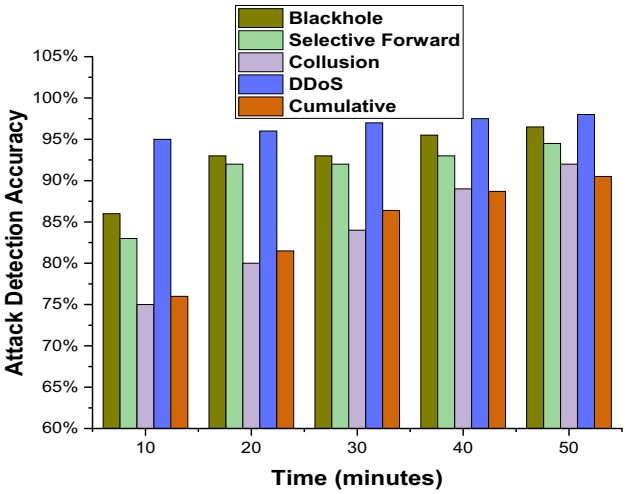


Fig. 16 Attack detection accuracy

when all the attacks are launched together, the cumulative attack detection accuracy remains more than 90%. Figure 17 compares the detection accuracies of GLSF^{2}IoT with various contemporaries, and it is observed that it gives far better results.

4.3 Performance of GLSF^{2}IoT in heterogeneous and scalable IoT

Heterogeneity is the classical characteristic of IoT. Diversity in software, hardware, and process requirements is justified by the range of functions performed by IoT devices. To show that the performance of GLSF^{2}IoT in detecting malicious nodes remains unaffected under heterogeneity and scalability, we implemented an IoT network (Fig. 11 scenario-3) initially with 60 edge nodes. Out of these 55 nodes, we configured 30 as sky-motes and the remaining 25 as Z1-Zolertia motes. For testing the scalability of GLSF^{2}IoT, the number of nodes was smoothly increased from 60 (20 attacker nodes; 12 at edge and 8 at fog) to 120 (40 attackers; 30 at the edge, 10 at the fog), and finally to 200 (85 attackers; 70 at the edge, 15 at the fog).

Sky-motes have 8 MHz Texas Instruments, MSP430 low power microcontroller, 10 KB RAM and 48 KB flash memory, 16-pin expansion support and optional SMA antenna connector [47], whereas Z1-Zolertia motes are equipped with 16 MHz MSP430F2617 low power microcontroller, 8 KB RAM, and 92 KB flash memory, 52-pin expansion support [48]. As such, there is heterogeneity in the capabilities and feature sets of the nodes.

With negligible (1.5%) variation in the detection accuracies of various attacks (Fig. 18), it is proven that GLSF^{2}IoT works efficiently for larger IoT networks and

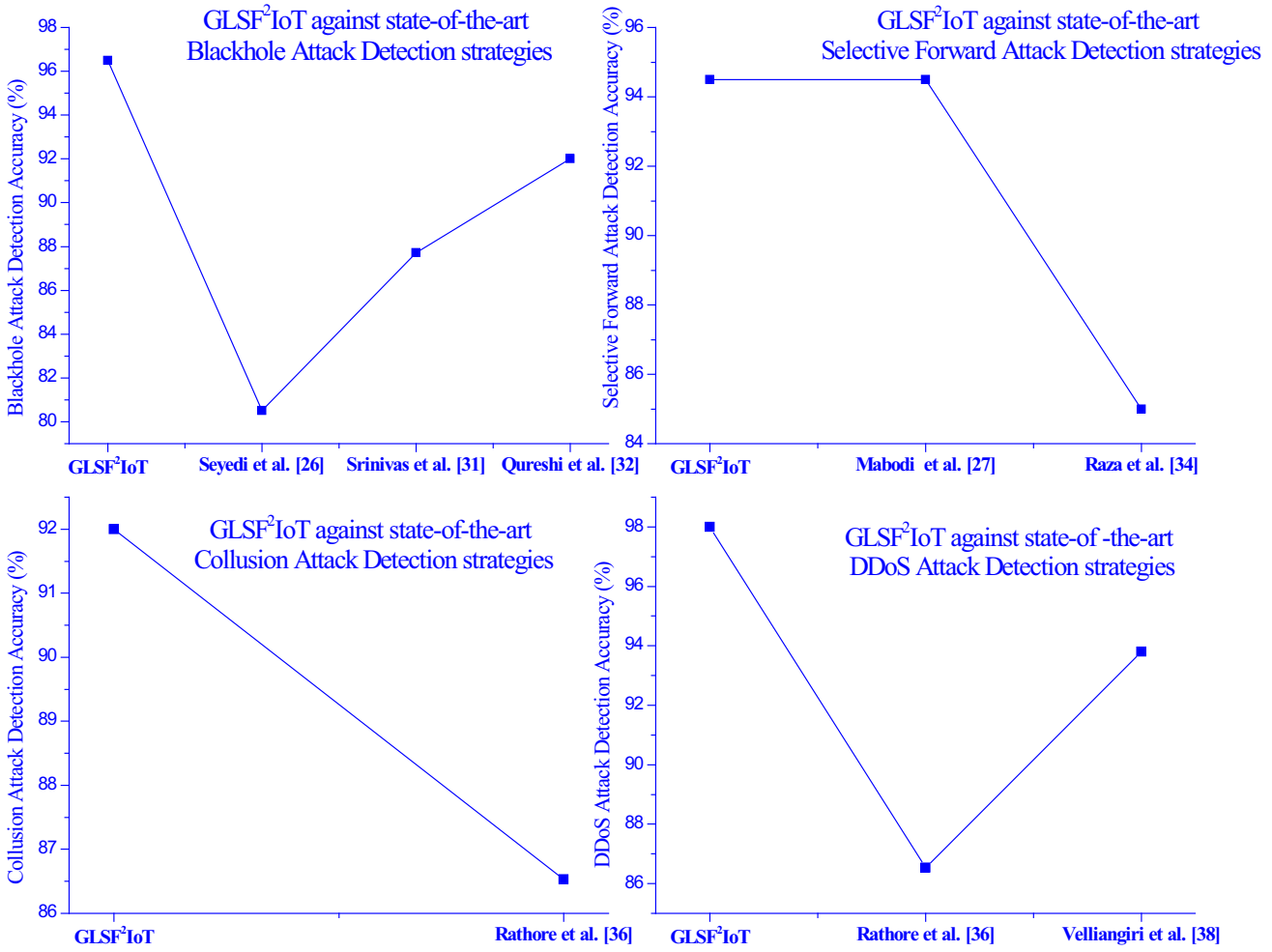


Fig. 17 GLSF2IoT against state-of-the-art attack detection strategies

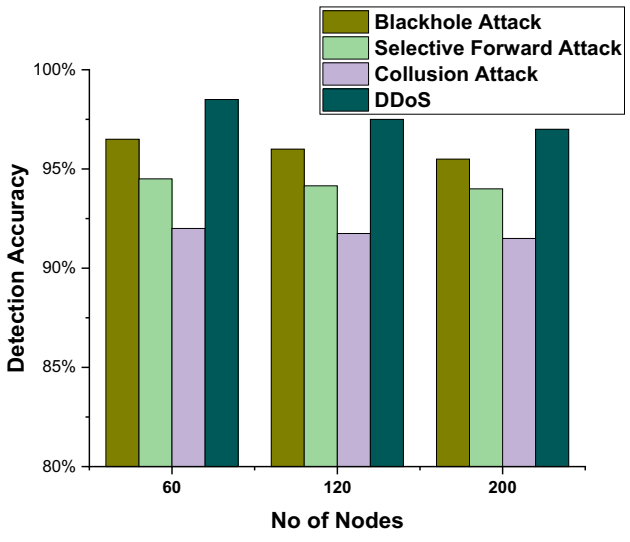


Fig. 18 Attack detection accuracy of GLSF2IoT in heterogeneous and scalable IoT networks

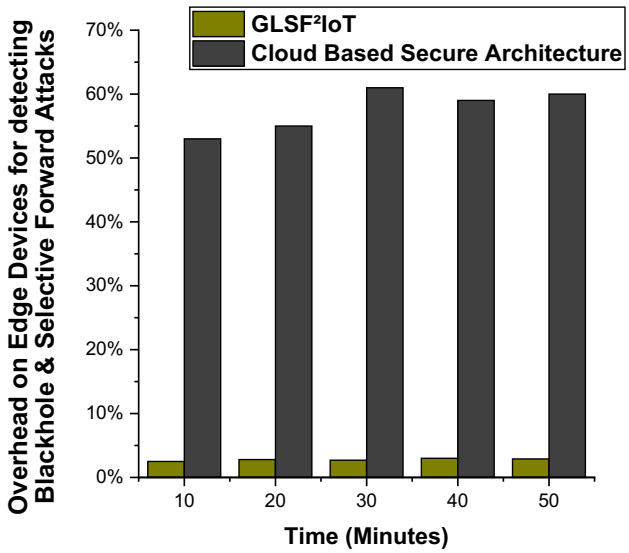


Fig. 19 Network overhead comparison between GLSF2IoT and cloud-based security architectures for detecting Blackhole and Selective forwarding attacks in IoT networks

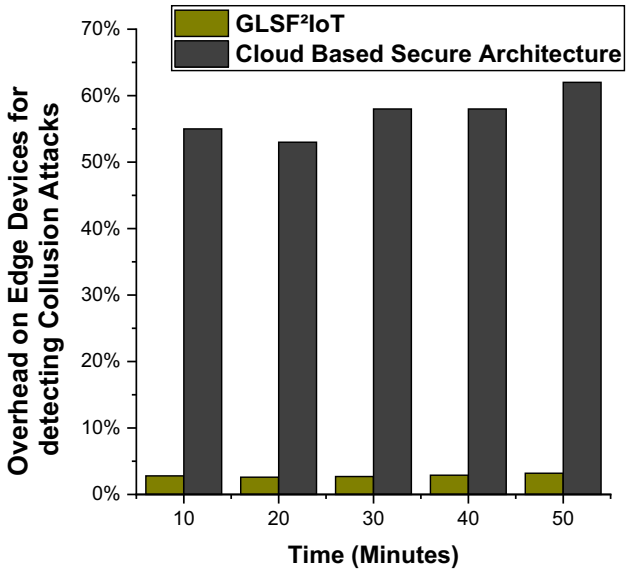


Fig. 20 Network overhead comparison between GLSF2IoT and cloud-based security architectures for detecting Collusion attack in IoT networks

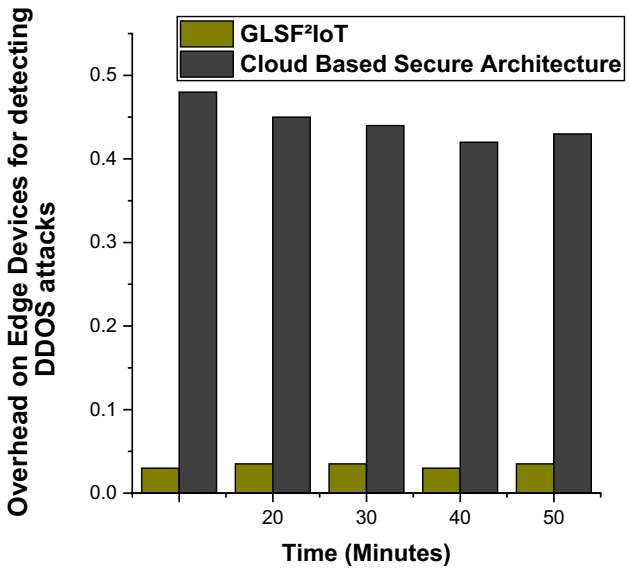


Fig. 21 Network overhead comparison between GLSF2IoT and cloud-based security architectures for detecting DDoS attack in IoT networks

performs the same for both heterogeneous and non-heterogeneous IoT networks, i.e., it is scalable and generic.

4.4 Performance comparison of GLSF^{2}IoT and cloud-based security architectures

The State-of-art techniques proposed by [31, 32, 34, 35] use cloud-based IoT architectures for providing security against various threats. Although the cloud provides massive storage and processing capabilities to IoT networks, it

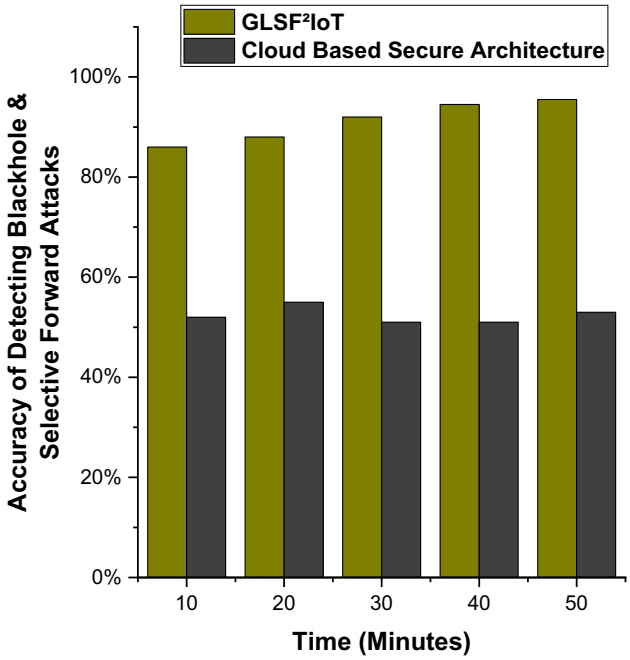


Fig. 22 Comparison of attack detection accuracy between GLSF2IoT and cloud-based security architectures for detecting blackhole and selective forwarding attacks in IoT networks

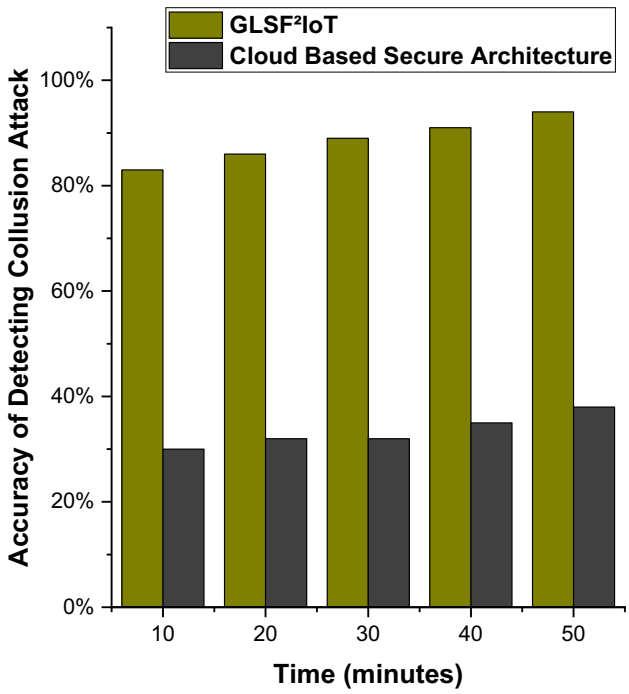


Fig. 23 Comparison of attack detection accuracy between GLSF2IoT and cloud-based security architectures for detecting Collusion attack in IoT networks

suffers from inherent disadvantages [49–51], viz. (a) being located at a distance from the edge IoT devices, (b) Huge bandwidth utilized for device-cloud communication, and

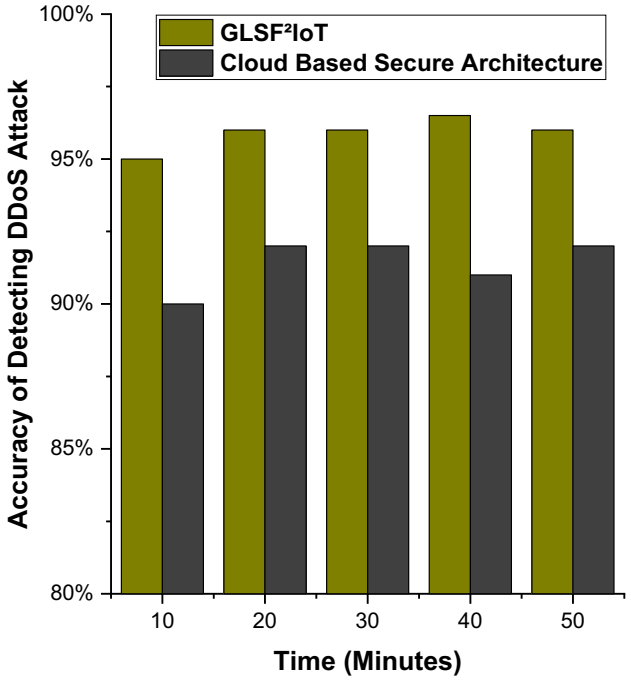


Fig. 24 Comparison of attack detection accuracy between GLSF2IoT and cloud-based security architectures for detecting DDoS attack in IoT networks

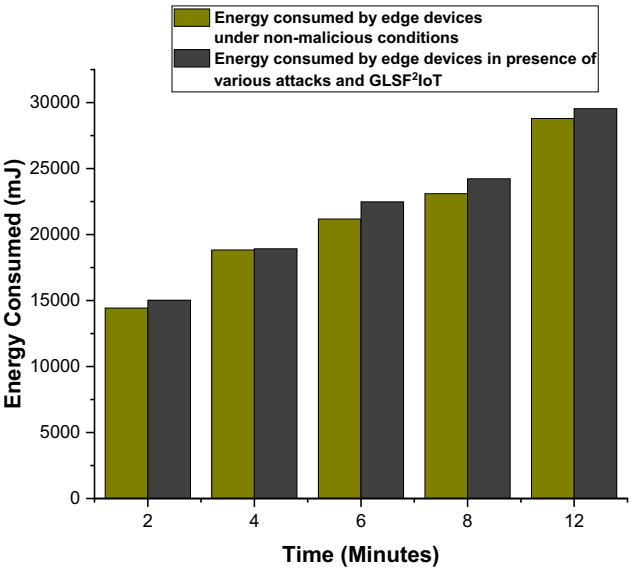


Fig. 25 Comparison of energy consumed by edge nodes in malicious (with GLSF2IoT) and non-malicious IoT network

(c) latency in furnishing the requested service. Fog-based IoT architecture, on the other hand, offers cloud services to the edge devices from the perimeter of the network, sharing the load of the network as all the devices don't need to connect directly with the cloud server [16].

To compare the two architectures, i.e., GLSF^{2}IoT and cloud-based security, we compare the overhead laid on the

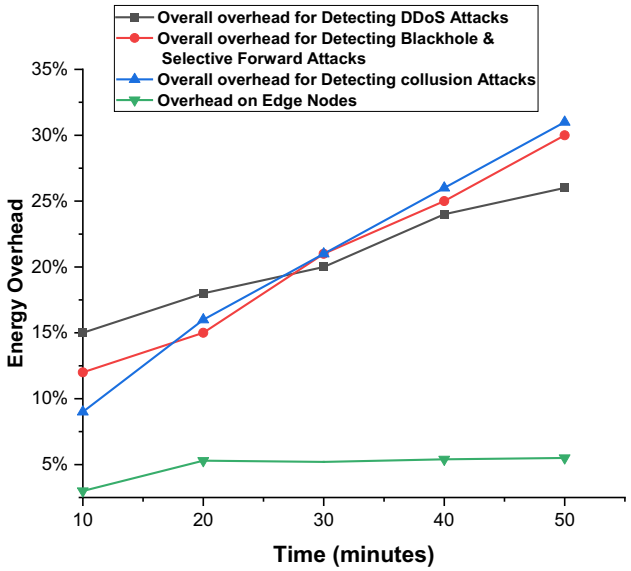


Fig. 26 Overall energy overhead for attack detection by GLSF2IoT

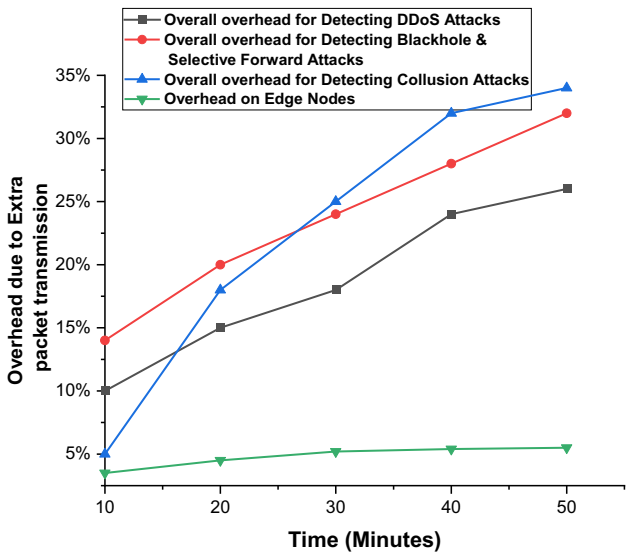


Fig. 27 Overall overhead due to extra packets transmitted by IoT nodes for attack detection using GLSF2IoT

edge devices. The overhead is calculated by analyzing the energy consumption of the edge nodes. This parameter is affected by the number of retransmissions required for establishing a connection with the cloud. The comparison is shown in Figs. 19, 20 and 21.

Figures 19, 20 and 21 orchestrate that the fog-based GLSF^{2}IoT security posture puts considerably less overhead on the edge devices than its cloud-based contemporaries [31, 32, 34, 35]. This is because all devices are directly connected to the cloud in cloud-based security designs. Because of the limited bandwidth, more energy is consumed for packet transmission and re-transmission attempts. The edge devices have to perform some security

Table 11 Memory overhead on various IoT network devices for implementing GLSF^{2}IoT

IoT device type	ROM overhead (in Bytes)	RAM overhead (in Bytes)	Overall memory overhead (%age)
Server node	4832	538–637	1.12
Fog node	3348	298–321	0.75
Edge device	213	138	0.072

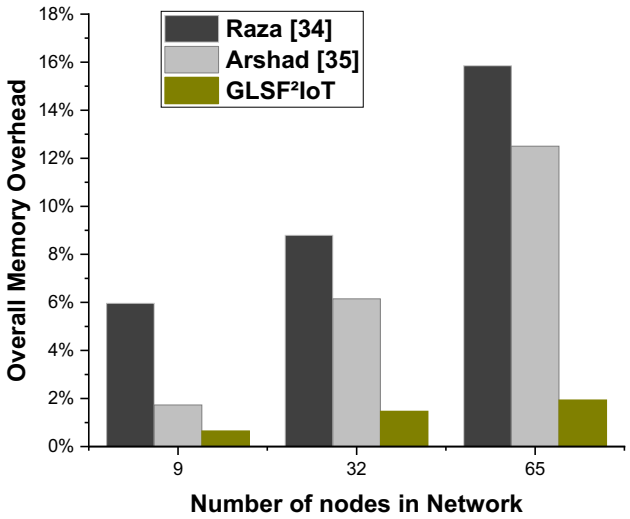


Fig. 28 Memory overhead comparison in implementing GLSF2IoT and other state-of-art security mechanisms to detect malicious attackers in IoT networks

functions themselves. In contrast, GLSF^{2}IoT shares the load of providing services and attack detection among multiple fog nodes, thereby sparing edge devices. As such, less number of retransmissions, and almost no pressure of attack detection is put on the edge nodes, which ultimately leads to less overhead.

Also, the number of devices being monitored by every single fog node is far less than the entire network monitored by a single cloud. Figures 22, 23 and 24 compare the detection accuracies of GLSF^{2}IoT and cloud-based security designs [31, 32, 34, 35]. It is seen that there is a massive gap between the accuracies of these architectures. Henceforth, fog-based architecture must be used for providing security to IoT environments where less latency, less bandwidth, and consideration to real-time analytics are needed.

4.5 GLSF^{2}IoT and energy overhead

To calculate the energy overhead on the edge devices in the presence of GLSF^{2}IoT security posture, we first calculated the average energy consumed by edge devices under normal conditions (scenario-2 of Fig. 11). Then we simulated scenario-3, instantiated GLSF^{2}IoT, and noted the average energy consumed by edge devices. The results are shown in Fig. 25. The energy value was calculated using the powertrace tool available in cooja. It gives the time taken by motes for transmission, reception, sleep, and processing of data. Using these values, we calculate the expression for energy as (Eq. 17):

$$E = \text{Voltage} \times (t_T \times 19.5 + t_R \times 21.8 + t_S \times 0.0545 + t_P \times 1.8) \quad (17)$$

And Power as (Eq. 18):

$$\text{Power(mW)} = {}^{E(\text{mJ})}/_{\text{Time}(s)} \quad (18)$$

where t_T, t_R are the times spent by motes for transmission and reception of packets, respectively, t_S is time spent during sleep mode or low power mode, and t_P is the time spent by mote during the processing of data. For checking the working conditions, i.e., voltage and current supplied in various modes of Tmote-sky motes, refer to [47]. Figure 25 shows the comparison of average energies consumed by edge IoT devices under normal and malicious environments.

Figure 25 shows a nominal variation in the average energy consumed by edge nodes while comparing a non-malicious IoT network with the malicious one. The overall energy and packet transmission overheads are depicted in Figs. 26 and 27. This proves that GLSF^{2}IoT is lightweight, i.e., for providing security, it doesn't rely on the already scarce energy of edge nodes.

Figures 26 and 27 indicate that the overall energy or extra packet transmission overhead in the presence of GLSF^{2}IoT on edge devices is 5%. Moreover, this overhead on cloud and fog nodes is not more than 30% of the energy consumed or other resources used by the devices.

4.6 GLSF^{2}IoT and memory overhead

IoT edge devices are constrained by storage space [52]. Therefore, the security posture designed for IoT must put less memory overhead on the edge devices. We calculated the overall memory overhead on network devices to implement GLSF^{2}IoT. It was seen that it uses 4.89 KB of

Table 12 Applicability of GLSF^{2}IoT to other cyber attacks

Type of attack	Can GLSF^{2}IoT tackle it?	Has GLSF^{2}IoT tackled it?	How has GLSF^{2}IoT tackled it?
Insider attacks	✔	✔	It honors the protocols of authentication and authorization but, it works on the principle of zero-trust, conducting an unceasing watch on the complete network to monitor both insider and outsider node activities
Data theft attacks	×	×	GLSF^{2}IoT was not developed on this level
Increased rank attack (attacker raises its rank to force its neighbors to choose another parent. Done to cause resource depletion and fragmentation in the network)	✔	✔	GLSF^{2}IoT constantly monitors rank inconsistencies in the network
Decreased rank attack (attacker advertises a smaller rank to attract more traffic. Done to disrupt the network traffic and launch attacks like blackhole, etc.)	✔	✔	GLSF^{2}IoT constantly monitors rank inconsistencies in the network
Worst Parent attack (attacker chooses the node with the highest rank for forwarding the packets of its children. These increase *count*, and delay in the network	✔	✔	GLSF^{2}IoT constantly monitors rank inconsistencies in the network
Impersonation (stolen credentials like username, passwords, etc.)	✔	✔(indirectly)	By doubting every node, GLSF^{2}IoT can identify this attacker even when s/he gains legal access to the otherwise secure network and becomes an insider
Compromised software (device's software is altered for launching the attack.)	✔	✔	For launching our chosen attacks, we have compromised the software of Tmote-sky motes
Compromised hardware	×	×	GLSF^{2}IoT was not developed on this level
Protocol deviation (attacker detours from the standard working of the protocol)	✔	×	Proper fuzzy rule-base needs to be created
Jamming attack (attacker blocks the channel and other resources by transmitting radio signals inappropriately)	✔	✔(indirectly)	As GLSF^{2}IoT continuously monitors the SNR, hiding of a jamming attacker is not possible
Eavesdropping (hearing the in-transit messages)	×	×	GLSF^{2}IoT was built on the active level. This is a passive attack
Tampering (altering, dropping or delaying the transmission)	Partly	Partly	If the attacker is modifying the contents of a message, GLSF^{2}IoT cannot identify the attack, but any other type of tampering will be identified
Man-in-the-middle (a corrupt node replaces a fog node and hijacks the secret information that was being shared)	✔	Partly (because the replacement has not been made)	GLSF^{2}IoT's fuzzy logic-based trust management mechanism keeps a watch on all the fog nodes. As and when it detects any malicious activity, it puts that node in the blacklist
Flooding (transport layer attack that drains the memory resources of its victim)	✔	✔	The legitimate PDR of a node drops when it forwards illegal packets. Since GLSF^{2}IoT continuously monitors PDR, it can detect this attack
De-synchronization (bars the end-points from achieving the sync by modifying sequence numbers)	✔	✔	By monitoring the ADSN, this attack can be identified as well
Overwhelming (flooding of application-layer traffic)	✔	✔(indirectly)	GLSF^{2}IoT keeps any kind of overwhelming under check

ROM for the implementation. From the detailed memory overhead shown in Table 11, it is evident that GLSF^{2}IoT puts negligible memory overhead on the devices. Figure 28 compares the memory overhead of GLSF^{2}IoT with those of two crucial state-of-art methods [34, 35].

The results in Fig. 28 are calculated by varying the number of IoT network devices and implementing

GLSF^{2}IoT and security mechanisms given in [34, 35] separately. This is done to resonate with the methods used for comparison. While [34] uses a maximum of 32 nodes, [35] creates a 15 node topology. To this end, we used a total of 9 nodes (1-server, 2- fog nodes, and 6-edge devices), 32 nodes (1-server, 4-fog nodes, and 27-edge devices), and 65 nodes (1-Server, 9-fog nodes, and 55-edge devices) for the first, second and third readings. In each scenario, we calculated the overall memory overhead for the mentioned security mechanism. Figure 28 indicates that GLSF^{2}IoT puts the least memory overhead than other benchmark security mechanisms.

5 Suitability of GLSF^{2}IoT for other IoT attacks

Although we have used the blackhole, selective forwarding, collusion, and DDoS attacks for validating the efficacy, and credibility of GLSF^{2}IoT, we do not in any way contend that only these attacks are worth tackling in an IoT set-up, but that they are dangerous, have high-impact, and can beat the purpose of any IoT network [25–27]. It is, however, critical that its validity against other attacks that could be launched in an IoT-fog set-up be pointed out. Table 12 reveals GLSF^{2}IoT's applicability to other IoT attacks.

6 Conclusion and future work

The paper analyzed the inevitability of digital annihilation without cyber-security. It was ascertained that if the sharing of data and information is needed beyond the "standalone" closed model, then security cannot be layered as an afterthought. As IoT brings more devices online, incorporating modern technologies into historically analog environments, increased security risks are preordained. This article studies the most dangerous attacks in the uncertain IoT environment and proposes GLSF^{2}IoT that incorporates the advantages of fog computing and fuzzy logic into the modeling of a lightweight security system for IoT. The work in this paper is the first that has dealt with the heterogeneity and uncertainty of the IoT environment. With the four procedures, and two layers working in tandem, it detects most of the attacks tossed in a constrained, heterogeneous, scalable and uncertain IoT landscape. Performance results affirm GLSF^{2}IoT 's optimality in attaining greater accuracy rates. Our work also helps to maintain the vital IoT component, i.e., people in the form of expert knowledge included in maintaining security. GLSF^{2}IoT operates in real-time, in contrast to the bulk of attack detection systems now available in the literature, which operate in post-attack mode.

In immediate future, we would like to attempt an extension to GLSF^{2}IoT for hardware compromise attacks. Efforts will be made to further reduce the false alarms, energy, and memory overheads in the overall network. Also, a ransomware attack in an IoT setting can be extremely dangerous given that the IoT devices have come very near to the personal lives of people. It can not only cause monetary loss, but lead to critical information breach and life risks. It has the potential to have an effect on the entire spectrum of security services, including transparency, confidentiality, and availability. For handling these ransomware attacks, we intend to use honeypot and Deep Learning-based method. A honeypot will lure the attacker to itself to analyze its modus operandi. The deep neural network will be trained to identify ransomware attacks and will be deployed on the honeypots. Consequently, we will be able to detect the ransomware attacks in real-time. However, given the extreme sophistication of cyber attackers today, detection of honeypots by them is also a huge challenge. To ensure the honeypots remain undetectable, we will investigate and evaluate potential strategies for detecting SSH and telnet honeypots. Also, we will check the applicability of unsupervised Deep Learning models and Reinforcement Learning to ransomware attack detection. Due to the intrinsic complexity of ransomware attacks, there are scenarios that cannot be characterized by a simple label. As a result, unsupervised Deep Learning-based approaches and Reinforcement Learning could perform well even if no prior knowledge of attack is available, which is an obvious advantage.

Declarations

Conflict of interest The author's declared that they have no conflict of interest.

References

1. Serror M, Hack S, Henze M, Schuba M, Wehrle K (2020) Challenges and opportunities in securing the industrial Internet of Things. IEEE Trans Ind Inf 17(5):2985–2996
2. Al-Garadi MA, Mohamed A, Al-Ali A, Du X, Ali I, Guizani M (2020) A survey of machine and deep learning methods for Internet of Things (IoT) security. IEEE Commun Surv Tutor 22(3):1646–1685
3. Zahra SR, Chishti MA (2019) Assesing the services, security threaths, challenges and solutions in the Internet of Things. Scal Comput: Pract Exp 20(3):457–484
4. Arbuckle A (2020) Addressing IoT device security head-on. https://www.securityweek.com/addressing-iot-device-security-head. Accessed 20 Nov 2020

5. National Law Review (2020) Buyer beware: the Internet of Things comes under new cyber attack from multiple fronts. https://www.natlawreview.com/article/buyer-beware-internet-things-comes-under-new-cyber-attack-multiple-fronts. Accessed 18 Dec 2020
6. Burke M (2020) Man hacks RING camera in 8-year-old girl's bedroom, taunts her: 'I'm Santa Claus'. https://www.nbcnews.com/news/us-news/man-hacks-ring-camera-8-year-old-girl-s-bedroom-n1100586. Accessed 20 Dec 2020
7. Hanrahan M (2020) Ring security camera hacks see homeowners subjected to racial abuse, ransom demands. https://abcnews.go.com/US/ring-security-camera-hacks-homeowners-subjected-racial-abuse/story?id=67679790#:~:text=Ring%20camera%20systems%20being%20hacked,-Multiple%20U.S.%20families&text=Owners%20of%20Ring%20security%20cameras,demanded%20a%20ransom%20in%20Bitcoin. Accessed 20 Dec 2020
8. Fier J (2020) Smart, or not so smart? What the ring hacks tell Us about the future of IoT. https://www.securityweek.com/smart-or-not-so-smart-what-ring-hacks-tell-us-about-future-iot. Accessed 21 December 2020
9. Haji S (2020) Essential IIoT security trends for 2020. https://www.securityweek.com/essential-iiot-security-trends-2020. Accessed 23 Dec 2020
10. Ballard B (2020) Millions of smart devices could still have major security flaws. https://www.techradar.com/in/news/millions-of-smart-devices-could-still-have-major-security-flaws. Accessed 26 Dec 2020
11. Holst A (2021) Global IoT end-user spending worldwide 2017–2025. https://www.statista.com/statistics/976313/global-iot-market-size/#:~:text=The%20global%20market%20for%20Internet,around%201.6%20trillion%20by%202025. Accessed 05 Jan 2021
12. Verified Market Research (2021) Internet of Things (IoT) Market worth $1319.08 Billion, Globally, by 2026 at 25.68% CAGR: verified market research. https://www.prnewswire.com/news-releases/internet-of-things-iot-market-worth-1319-08-billion-globally-by-2026-at-25-68-cagr-verified-market-research-301092982.html. Accessed 06 January 2021
13. Matthews K (2021) What do IoT hacks cost the economy? https://www.iotforall.com/iot-hacks-cost#:~:text=Attacks%20Damage%20Revenue&text=The%20survey%20polled%20approximately%20400,13.4%20percent%20of%20annual%20revenue. Accessed 08 Jan 2021
14. Kleinman L (2021) Attack from DOS: in zero we trust. https://securitybrief.co.nz/story/attack-from-dos-in-zero-we-trust. Accessed 10 Jan 2021
15. Ponemon Institute (2021) 2018 Cost of insider threats: global. https://www.insiderthreatdefense.us/pdf/Ponemon%20Institute%202018%20Report%20-%20The%20True%20Cost%20Of%20Insider%20Threats%20Revealed.pdf. Accessed 12 Jan 2021
16. Zahra SR, Chishti MA (2020) Fuzzy logic and fog based secure architecture for Internet of Things (FLFSIoT). J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02128-2
17. Zadeh LA (1975) Fuzzy logic and approximate reasoning. Synthese 30(3–4):407–428
18. SOS children's villages Canada (2020) Poverty in India: two-third of people are considered extremely poor. https://www.soschildrensvillages.ca/news/poverty-in-india602#:~:text=Two%2Dthirds%20of%20people%20in,they%20are%20considered%20extremely%20poor. Accessed 22 Dec 2020
19. Zadeh LA (1988) Fuzzy logic. Computer 21(4):83–93
20. TM Forum (2020) 70 percent of IoT devices 'vulnerable to attack'. https://inform.tmforum.org/news/2014/07/70-percent-iot-devices-vulnerable-attack/. Accessed 27 Dec 2020
21. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
22. Dzitac I, Filip FG, Manolescu MJ (2017) Fuzzy logic is not fuzzy: world-renowned computer scientist Lotfi A. Zadeh. Int J Comput Commun Control 12(6):748–89
23. Zadeh LA (1999) From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. IEEE Trans Circuits Syst I: Fundam Theor Appl 46(1):105–19
24. Zadeh LA (2001) A new direction in AI: toward a computational theory of perceptions. AI Mag 22(1):73–73
25. Mathur A, Newe T, Rao M (2016) Defence against black hole and selective forwarding attacks for medical WSNs in the IoT. Sensors 16(1):118
26. Seyedi B, Fotohi R (2020) NIASHPT: a novel intelligent agent-based strategy using hello packet table (HPT) function for trust Internet of Things. The J Supercomput 76(9):1–24
27. Mabodi K, Yusefi M, Zandiyan S, Irankhah L, Fotohi R (2020) Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication. J Supercomput: 1–26.
28. Vijayakumar P, Chang V, Deborah LJ, Balusamy B, Shynu PG (2018) Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks. Futur Gener Comput Syst 78:943–955
29. Yaseen Q, Jararweh Y, Al-Ayyoub M, AlDwairi M (2017) Collusion attacks in internet of things: detection and mitigation using a fog based model. In: 2017 IEEE sensors applications symposium (SAS). IEEE. pp. 1–5
30. Ouechtati H, Azzouna NB, Said LB (2019) A fuzzy logic based trust-ABAC model for the Internet of Things. In: International conference on advanced information networking and applications. Springer, Cham. pp. 1157–1168
31. Srinivas TA, Manivannan SM (2020) Preventing collaborative black hole attack in IoT construction using a CBHA–AODV routing protocol. Int J Grid High Perform Comput (IJGHPC) 12(2):25–46
32. Qureshi KN, Rana SS, Ahmed A, Jeon G (2020) A novel and secure attacks detection framework for smart cities industrial internet of things. Sustain Cities Soc 61:102343
33. Ribera EG, Alvarez BM, Samuel C, Ioulianou PP, Vassilakis VG (2020) Heartbeat-based detection of blackhole and greyhole attacks in RPL networks. In: 2020 12th international symposium on communication systems, networks and digital signal processing (CSNDSP). IEEE. pp. 1–6
34. Raza S, Wallgren L, Voigt T (2013) SVELTE: real-time intrusion detection in the Internet of Things. Ad Hoc Netw 11(8):2661–2674
35. Arshad J, Azad MA, Abdeltaif MM, Salah K (2020) An intrusion detection framework for energy constrained IoT devices. Mech Syst Sign Process 136:106436
36. Rathore S, Park JH (2018) Semi-supervised learning based distributed attack detection framework for IoT. Appl Soft Comput 72:79–89
37. Haripriya AP, Kulothungan K (2019) Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. EURASIP J Wirel Commun Netw 2019(1):90
38. Velliangiri S, Pandey HM (2020) Fuzzy-Taylor-elephant herd optimization inspired deep belief network for DDoS attack detection and comparison with state-of-the-arts algorithms. Fut Gener Comput Syst 110:80–90
39. Yang Y, Zheng X, Liu X, Zhong S, Chang V (2018) Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system. Futur Gener Comput Syst 84:160–176
40. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805

41. Smith B (2020) A moment of reckoning: the need for a strong and global cybersecurity response. https://blogs.microsoft.com/on-the-issues/2020/12/17/cyberattacks-cybersecurity-solarwinds-fireeye/. Accessed 28 Dec 2020
42. Yi J, Kim S, Kim J, Choi S (2020) Supremo: cloud-assisted low-latency super-resolution in mobile devices. IEEE Trans Mobile Comput. https://doi.org/10.1109/TMC.2020.3025300
43. Kamgueu PO, Nataf E, Djotio TN (2015) On design and deployment of fuzzy-based metric for routing in low-power and lossy networks. In: 2015 IEEE 40th local computer networks conference workshops (LCN Workshops). IEEE. pp. 789–795
44. Moudni H, Er-rouidi M, Mouncif H, El Hadadi B (2019) Black hole attack detection using fuzzy based intrusion detection systems in MANET. Procedia Comput Sci 151:1176–1181
45. Khalil I, Bagchi S (2010) Stealthy attacks in wireless ad hoc networks: detection and countermeasure. IEEE Trans Mob Comput 10(8):1096–1112
46. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. Futur Gener Comput Syst 82:761–768
47. Moteiv (2020) Tmote Sky. https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf. Accessed 15 Nov 2020
48. Advancare SL (2020) Zolertia. http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf. Accessed 17 November 2020
49. Sabireen H, Neelanarayanan V (2021) A review on fog computing: architecture, fog with IoT. Algoritm Res Chall ICT Expr 7(2):162–176
50. Habibi P, Farhoudi M, Kazemian S, Khorsandi S, Leon-Garcia A (2020) Fog computing: a comprehensive architectural survey. IEEE Access 8:69105–69133
51. Ijaz M, Li G, Lin L, Cheikhrouhou O, Hamam H, Noor A (2021) Integration and applications of fog computing and cloud computing based on the Internet of Things for provision of healthcare services at home. Electronics 10(9):1077
52. Cao K, Liu Y, Meng G, Sun Q (2020) An overview on edge computing research. IEEE Access 8:85714–85728