Greater impact

2.5x more Altmetric attention. OA articles attracted 1.9x more news mentions and 1.2x more policy mentions

Find out more about benefits of open access.

Fees and Funding

Article processing charges (APC)

Authors who publish open access in *SN Computer Science* are required to pay an article processing charge (APC). The APC price will be determined from the date on which the article is accepted for publication.

The current APC, subject to VAT or local taxes where applicable, is:

£2190.00/$3090.00/€2490.00

Visit our open access support portal and our Journal Pricing FAQs for further information.

Authors can also choose to publish under the traditional publishing model (no APC charges apply); both options will be offered after the paper has been accepted.

Open access funding

Visit Springer Nature's open access funding & support services for information about research funders and institutions that provide funding for APCs.

Springer Nature offers agreements that enable institutions to cover open access publishing costs. Learn more about our open access agreements to check your eligibility and discover whether this journal is included.

Marcelo Zambrano

De: MIGUEL ANGEL BOTTO TOBAR <mbotto@gdeon.org>
Enviado el: jueves, 22 de agosto de 2024 9:06
Para: Marcelo Zambrano
Asunto: Your Submission SNCS-D-24-02228R1 - [EMID:fa82f6b1492957fc]

Dear Dr. Zambrano-Vizuete,

We are pleased to inform you that your manuscript, "Evaluating the Sustainability of Cerebral Edge Computing Inventiveness for Acquiring Internet of Things Substructure Autonomously", has been accepted for publica□on in **SN Computer Science**.

Please remember to quote the manuscript number, **SNCS-D-24-02228R1**, whenever inquiring about your manuscript.

Before starting the process of publishing your manuscript it is necessary to pay the production fee. Afterwards, you will receive an email regarding the production time and process. The cost of publishing a manuscript in SN Computer Science is $3090.00.

https://link.springer.com/journal/42979/how-to-publish-with-us#Fees%20and%20Funding

Due to your collaboration as a reviewer in our journal and the current agreement between GDEON S.A. and the Springer — International Publisher, you are entitled to a 50% discount for your publication, that is, the publication cost will be $1545.00.

Payment information:

Account holder:	GDEON S.A.
RUC:	1291779502001
Address:	Cdla. Los Laureles – Av Principal s/n vía a Montalvo, Guayaquil-Ecuador
Subject:	Publication of manuscript ID: SNCS-D-24-02228R1
Bank:	Pichincha
Current Account:	2100203261

With kind regards,

Lead Guest Editor
SN Computer Science

Comments to the author (if any):

Reviewer #2: The necessary changes are made.

The manuscript looks good.

Authors may need to take specific actions to achieve compliance with funder and institutional open access mandates. If your research is supported by a funder that requires immediate open access (e.g. according to Plan S principles) then you should select the gold OA route, and we will direct you to the compliant route where possible.

For authors selecting the subscription publication route our standard licensing terms will need to be accepted, including our self-archiving policies. Those standard licensing terms will supersede any other terms that the author or any third party may assert apply to any version of the manuscript.

 Find out more about compliance: https://www.springernature.com/gp/open-research/funding/policy-compliance-faqs.

This letter contains confidential information, is for your own use, and should not be forwarded to third parties.

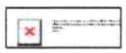
Recipients of this email are registered users within the Editorial Manager database for this journal. We will keep your information on file to use in the process of submitting, evaluating and publishing a manuscript. For more information on how we use your personal details please see our privacy policy at https://www.springernature.com/productionprivacy-policy.

If you no longer wish to receive messages from this journal or you have questions regarding database management, please contact the Publica□on Office at the link below.

In compliance with data protection regulations, you may request that we remove your personal registration details at

any time. (Use the following URL: https://www.editorialmanager.com/sncs/login.asp?a=r). Please contact the publication office if you have any questions.

--



Miguel Botto-Tobar
GDEON S.A.
mbotto@gdeon.org
https://www.gdeon.org

SN Computer Science (2024) 5:922
https://doi.org/10.1007/s42979-024-03220-6

ORIGINAL RESEARCH

Evaluating the Sustainability of Cerebral Edge Computing Inventiveness for Acquiring Internet of Things Substructure Autonomously

Marcelo Zambrano-Vizuete[1] · Juan Minango-Negrete[1] · Wladimir Paredes-Parada[2] · Jorge Pérez-Chimborazo[1] · Ana Zambrano-Vizuete[3]

Received: 5 April 2024 / Accepted: 11 August 2024

Abstract

Advancements in IoT have integrated it into every aspect of human life. By using the Internet as its foundation, IoT connects a vast array of cyber-physical devices, from simple sensors to advanced servers. However, this extensive connectivity also broadens the attack surface, increasing vulnerability to cyber threats due to the complex communication and non-standard technologies involved. The proposed response selection method addresses this by employing fuzzy logic inference at edge nodes, which processes the ambiguous data generated by IoT devices. The system evaluates four metrics: device importance, severity score, response cost, and success rate, calculated efficiently at the edge. Proximity to end devices makes edge nodes ideal for this task. Simulations reveal that the edge-based approach improves response success rates by about 5% compared to cloud-based implementations. This underscores the model's ability to accurately and swiftly select appropriate responses, demonstrating the effectiveness of edge computing in enhancing IoT security and performance.

Keywords IoT · EDGE · Fuzzy · Cyber-attack · Cloud computing

Introduction

Edge computing allows for dispersed compute, storage, networking, and communication functionality closer to the end devices in an IoT context, thereby bridging the gap between the cloud and end devices. Edge computing helps ensure the cyber security of the Internet of Things, in addition to the previously mentioned benefits of lower bandwidth use and latency [1]. Edge nodes, which operate in the continuum from the cloud to the devices, boost IoT security. By moving compute, storage, and networking closer to end devices, edge computing improves security [2].

Edge computing shifts the processing and management of data from distant data centres to edge nodes closer to the end devices. The edge nodes must be traversed by all data travelling between the cloud and the end devices. This allows the edge nodes to swiftly identify attacks and counter them [3].

High availability, security, and privacy are all supported by storing data from end devices in the distributed

✉ Marcelo Zambrano-Vizuete
marcelo.zambrano@ister.edu.ec

Juan Minango-Negrete
juancarlos.minango@ister.edu.ec

Wladimir Paredes-Parada
wladimir.paredes@ister.edu.ec

Jorge Pérez-Chimborazo
jorge.perez@ister.edu.ec

Ana Zambrano-Vizuete
ana.zambrano@epn.edu.ec

[1] Instituto Tecnológico Superior Rumiñahui, Sangolquí, Ecuador

[2] Consejo de Aseguramiento de la Calidad de la Educación Superior, Instituto Tecnológico Superior Rumiñahui, Quito, Ecuador

[3] Escuela Politécnica Nacional, Quito, Ecuador

Published online: 30 September 2024

edge nodes that are physically closer to the user's premise. Edge computing allows for end-to-end communication to take place closer to the user, protecting their privacy and making eavesdropping much less likely [4].

The IoT devices that are low on resources and therefore unable to run complex security protocols and algorithms are safeguarded by edge computing as shown in Fig. 1. Edge computing's decentralised, multi-tiered design offers numerous layers of protection for end devices, allowing for defence in depth. Credentials for end-device security and software upgrades can be stored in the "Edge," thanks to edge computing [5]. Edge computing's distributed architecture allows for reliable security status monitoring. Edge computing's efficient distributed processing is great, but IoT also need an autonomic security strategy. The reason for this is that Internet of Things devices can be found in both managed and unmanaged settings. Devices in an unsupervised setting are easier targets for hackers. In order to foresee, detect, and defend against cyber-attacks with minimal to no human interaction, security technique in IoT can apply the self-protect capabilities of autonomic computing [6].

A self-protection system is an autonomous network that can safeguard its own resources from intrusion and ensure the system's continued viability and safety. Autonomic computing, which is necessary for Internet of Things applications, has a self-protection feature. In the ever-changing IoT ecosystem, system integrity is dependent on self-protection features that guard against compromise [7]. There are two ways to approach the self-protection features: reactively and proactively. If the system is to be considered reactive, it must be able to identify attacks on its own and take appropriate action in response. To be deemed proactive, a system must predict when an attack is likely to occur and take steps to prevent it immediately. A self-protecting system must independently recognise aberrant behaviour and deploy its own defence mechanisms in response to threats [8]. The self-protection system keeps an eye out for any suspicious behaviour, and then takes the necessary measures to deal with any potential attacks.

In order to design an IoT self-protection system, it is necessary to extract intelligence from a massive amount of varied data, which is coming in at a rapid pace from a wide variety of heterogeneous IoT devices. To meet this need, self-protection systems in the IoT make use of machine learning techniques. IoT's decentralised architecture necessitates a distributed security mechanism that allows for diverse devices to communicate with one another and scale with minimal disruption [9]. In addition, IoT devices are used in extremely important applications, and they are not only the targets of assaults but also the backbones of malicious botnets. Consequently, the self-protection system must provide prompt and effective decisions from the copious streams of real-time data produced by the Internet of Things. This need is met by edge computing, and its dispersed intelligence may be used by the self-protection system by moving computations and data from IoT devices to edge nodes rather than a cloud server, allowing for more rapid detection and defence against threats [10–12].

Due to its numerous vulnerabilities, IoT infrastructure must prioritize security above all else. First, the number of Internet of Things devices is growing exponentially, found in both managed and unmanaged settings, creating more entry points for attackers. Remote, unmanaged devices can serve as launch points for attacks, and compromised gadgets within the user's physical environment can inflict direct harm. Second, the reliance on a wide variety of devices and technologies leads to interoperability issues, which can be exploited by malicious actors. Each new device and technology integration point presents a potential vulnerability that attackers can target. Third, IoT's use of wireless technologies increases its

Fig. 1 IoT security challenges

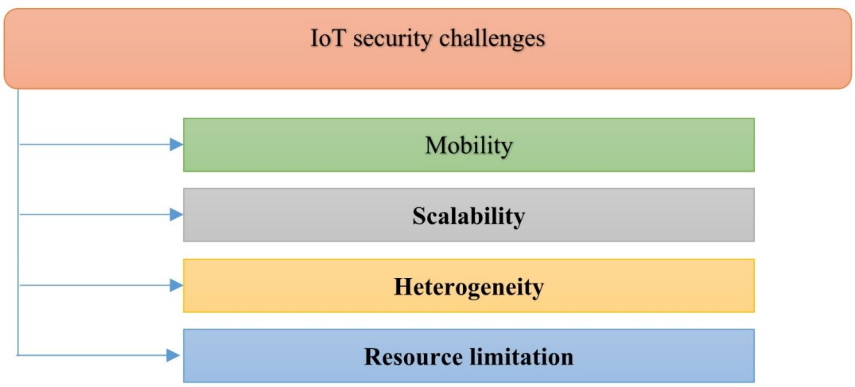


Table 1 Comparison of the different intrusion detection systems (IDS)

IDS Approach	Description	Detection Technique	Pros	Cons	Relevant Protocols/Settings
Hybrid IDS for IP-connected sensor networks [14]	Combines signature- and anomaly-based detection for low-power sensors	Distributed pattern-matching algorithm at sensor nodes	Efficient for low-power sensors	Early stage, limited research in IoT	General IoT sensor networks
Specification-based IDS for RPL topology [15]	Detects RPL topology attacks like rank and local repair assaults	Finite state machine design	Prevents contemporary attacks	Requires manual building of normal behavior knowledge	RPL-based IoT networks
Hybrid IDS for 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) using RPL [16]	Centralized and distributed design to identify various attacks	Centralized module 6Mapper at 6BR maps RPL topology and IDS settings	Effective in identifying DoS, rank attacks, and RPL inconsistencies	High energy consumption for frequent topology information updates	6LoWPAN, RPL protocol
Signature-based IDS for DoS attacks in ebbits network [17–19]	Architecture based on EU FP7 project ebbits	Signature matching algorithms	Specific to ebbits network, detects DoS attacks	Not compatible with other protocols	ebbits network using 6LoWPAN
Intrusion detection using Complex Event Processing [20]	Real-time pattern matching of IoT events to detect intrusions	Complex Event Processing	Reduces false alarms	High resource usage, impractical for many IoT devices	General IoT devices
Wormhole attack detection in 6LoWPAN [21]	Detects wormhole attacks using RPL DODAG node locations and signal intensity	Analysis of DODAG node and neighbor locations	Effective for non-leaf nodes	Limited to 6LoWPAN using RPL, specific to non-leaf node attacks	6LoWPAN, RPL protocol
Game theory-based IDS for IoT [22, 23]	Combines signature and anomaly analysis, uses Nash Equilibrium	Game model for authorized users and invaders	Supports low power IoT devices	Cannot analyze all forms of attacks in dynamic IoT environments	General IoT settings
Anomaly detection at the edge [24–26]	Uses hyper-ellipsoidal clustering algorithm for detecting outliers	Edge-enabled detection systems	High accuracy and timely detection	Lacks built-in support for on-the-fly learning	General IoT devices
Deep learning and edge computing-based IDS [27]	Distributed edge nodes train the model; coordinating node updates global model	Deep learning with client-server model	Leverages edge computing	Time-consuming training and updating process	General IoT networks
Edge computing with OS-ELM [28–30]	Utilizes Online-Sequential Extreme Learning Machine for edge network intrusion detection	Online learning of cyber-attacks	Swift identification, incorporates new data for learning	New approach, specific to thesis	General IoT settings

susceptibility to attacks such as inference and eavesdropping, where unauthorized parties intercept and manipulate data transmissions. These wireless connections, while convenient, often lack robust security measures, making them prime targets for cybercriminals [13].

Given these security challenges, it is crucial to implement comprehensive protective measures. This includes employing encryption to safeguard data transmissions, ensuring regular software updates to patch vulnerabilities, and utilizing strong authentication protocols to prevent unauthorized access. Additionally, network segmentation can limit the potential damage from any single compromised device by isolating it from other networked devices. By prioritizing security and addressing these vulnerabilities, we can better protect IoT infrastructure from the wide array of potential cyber threats it faces. In conclusion, the rapid expansion of IoT devices and their inherent vulnerabilities demand a proactive and multi-faceted approach to security. This ensures that as IoT continues to grow and integrate into various aspects of life, it remains secure and resilient against evolving cyber threats.

This study introduces a novel multi-layered security framework for IoT environments, integrating lightweight cryptographic algorithms with real-time threat detection and adaptive security protocols. By addressing the unique vulnerabilities and communication challenges of IoT devices, this approach offers a tailored, robust solution not feasible with traditional security methods.

The rest of the paper is structured as follows. The related work is briefly described in part 2, and the motivation for the research work is discussed in Sect. 3 and methodology and the theoretical foundations of the methods used are described in Sect. 4. The simulation results and analysis are presented in Sect. 5. We summarize the most important results and key findings in conclusion section.

Existing Work Done

While there has been a lot of work done on intrusion detection systems, their use in IoT is still in its early stages. There are currently very few research dealing with intrusion detection in the Internet of Things. Here's a tabular comparison of the different intrusion detection systems (IDS) discussed in the content:

This Table 1 summarizes the key aspects of each IDS approach, including their description, detection technique, advantages, disadvantages, and applicable protocols or settings.

In many ways, the intrusion detection system developed in this thesis is distinct from these other publications. To begin with, it employs edge computing to swiftly identify an assault. In addition, OS-ELM, an Online-Sequential Extreme Learning Machine, is deployed across the edge network to identify intrusions. Finally, the suggested detection system has a major advantage over the compared algorithms in that it can include new data in online for learning the cyber-attacks.

Motivation for the Research Work

Attackers now have a larger attack surface to work with because to the increased connectivity and heterogeneity of IoT. If a cyber-attack on a vital infrastructure causes a catastrophe or human casualties, it is imperative that the attack be detected as soon as possible in order to stop it or mitigate its effects by triggering appropriate countermeasures. As a result, the Internet of Things needs a sophisticated security system that can deduce the nature of attacks from IoT traffic and respond appropriately and quickly. By moving computations and data from IoT devices to the edge nodes instead of the cloud server, self-protection systems in IoT can exploit the dispersed intelligence of edge computing to identify and recover from cyber-attacks more quickly.

Fuzzy logic's capacity to handle imprecise and partial data, as well as approximation reasoning for decision making, has made it a popular tool in many IoT applications. Decisions made using fuzzy logic have been shown to be accurate. To make decisions based on the approximate data collected from IoT devices during an attack, fuzzy logic is employed.

The Proposed Work

The exponential growth in the number of Internet of Things (IoT) devices is a major element in driving the requirement for self-protect features. Therefore, it is up to the devices to employ their own inherent security measures in the face of a cyber attack. The attack can be detected and countered by the self-protect feature with little help from a person. However, IoT devices might be anything from simple sensors to highly computational systems, making it impossible to use sophisticated security techniques and algorithms.

The value of an IoT device is determined using the standard weighted average formula. The above criteria serve as weights for the aforementioned attribute x of IoT devices.

$$w(x) = \begin{cases} 0.1, \ if \ 0 \leq \ x \leq \ t_1 \\ 0.5, \ if \ t_1 \leq \ x \leq \ t_2 \\ 1, \ if \ t_2 \leq \ x \leq \ 1 \end{cases} \quad (1)$$

Where, t_1 and t_2 are thresholds used to assign values to the properties of the device being weighed. The minimum criticality value can be specified with a weight of 0.1, a moderate criticality value with 0.5, and a high criticality value with 1. Data created by a sensor used in critical health care monitoring, for instance, has a higher value than data generated by the same sort of sensor used in atmospheric monitoring, and these thresholds are determined based on the IoT application in which the devices are used.

The majority of IoT apps store and process information gathered from end devices on the cloud. Offloading processing and increasing delay are two aspects of cloud-based security approaches. In order to keep Internet of Things devices safe from intrusion, the authors of this research suggest implementing an edge computing–based self–protection mechanism. Figure 2 shows how the proposed system was developed. When an attack is detected on an IoT device, this system uses that information as input to determine what kind of response is needed to either halt the assault or mitigate its effects. Distributed edge nodes, which are physically closer to end devices, run the self-protection system. Since edge nodes are located nearer to end devices, their self-protection systems can detect attacks and launch countermeasures more quickly than those in the cloud described in Fig. 2.

Cloud-based approaches for IoT security suffer from significant drawbacks, including processing latency due to data transmission delays, increased bandwidth usage, and potential single points of failure. These issues can

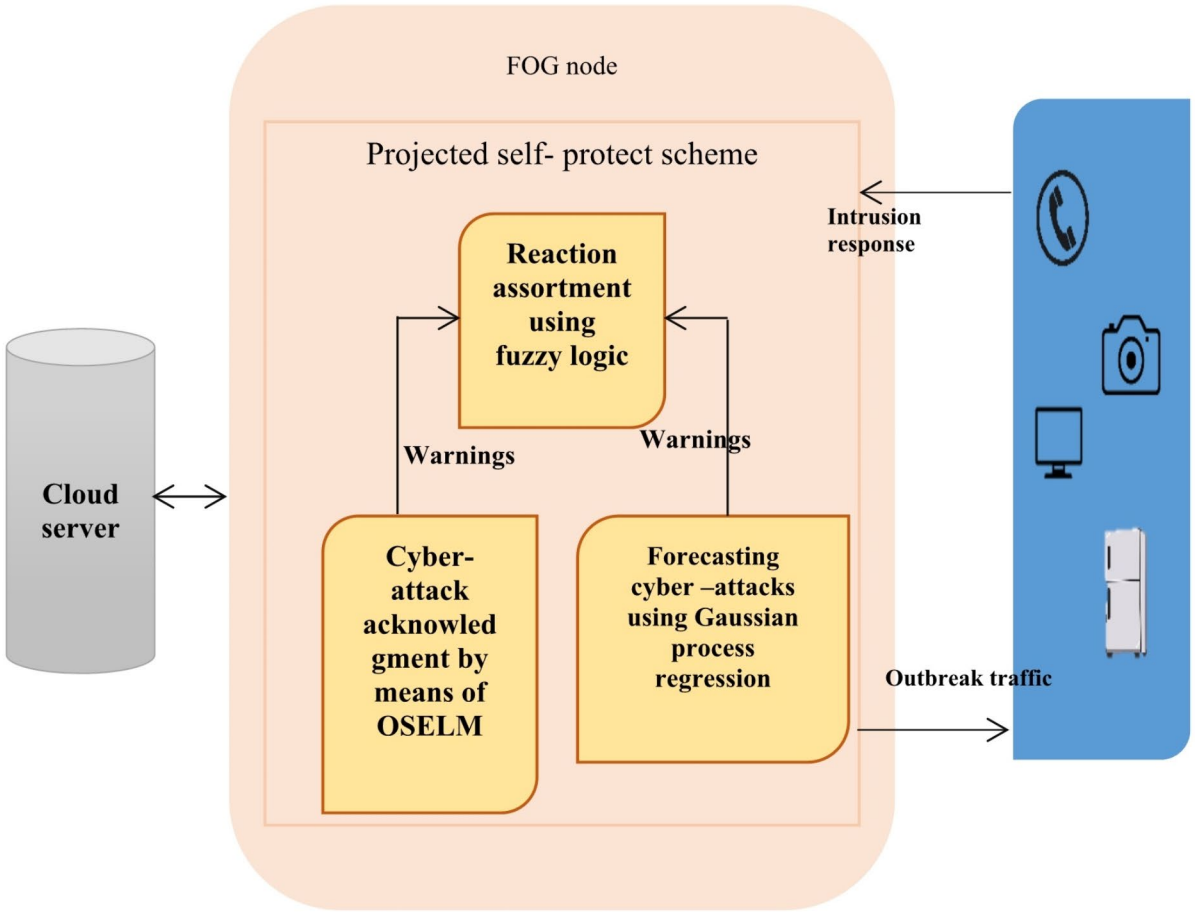


Fig. 2 Enterprise procedure of projected self-protection scheme

hinder real-time threat detection and response, compromising the efficiency and reliability of IoT systems.

As an initial step, the proposed Online-Sequential Extreme Learning Machine (OSELM) based intrusion detection approach analyzes the input traffic from the IoT environment to identify known attack patterns. OSELM is utilized in this detection mechanism to facilitate rapid learning in parallel at remote edge nodes from sequentially provided streaming data in an IoT setting. OSELM's strong generalization capabilities enable it to learn quickly and effectively from the dynamic streaming data characteristic of IoT environments. Additionally, the proposed cyber-attack forecasting system employs distributed Gaussian process regression to analyze IoT traffic and detect zero-day attacks, which do not conform to established attack patterns. This forecasting method leverages distributed Gaussian process regression to make accurate predictions from large datasets with minimal training time, even in the context of IoT-generated data. Figure 3 details the Projected fuzzy inference scheme for reaction assortment.

Finally, a proposed response mechanism is introduced, which selects the appropriate reaction based on the characteristics and dynamic nature of the IoT device. This mechanism uses fuzzy logic inference at the edge nodes to interpret and act on the uncertain data typical of an IoT environment.

In this section, we explore the criteria for evaluating the effectiveness of edge computing solutions. Key factors include latency, bandwidth usage, user engagement, energy efficiency, and cost-effectiveness. Together, these metrics offer a comprehensive assessment of the proposed strategies and algorithms. To provide a quantitative understanding of the impact of edge computing on broadcasting, these same metrics should be used for comparing different edge computing systems.

By focusing on these evaluation standards, we can better understand how edge computing solutions perform in

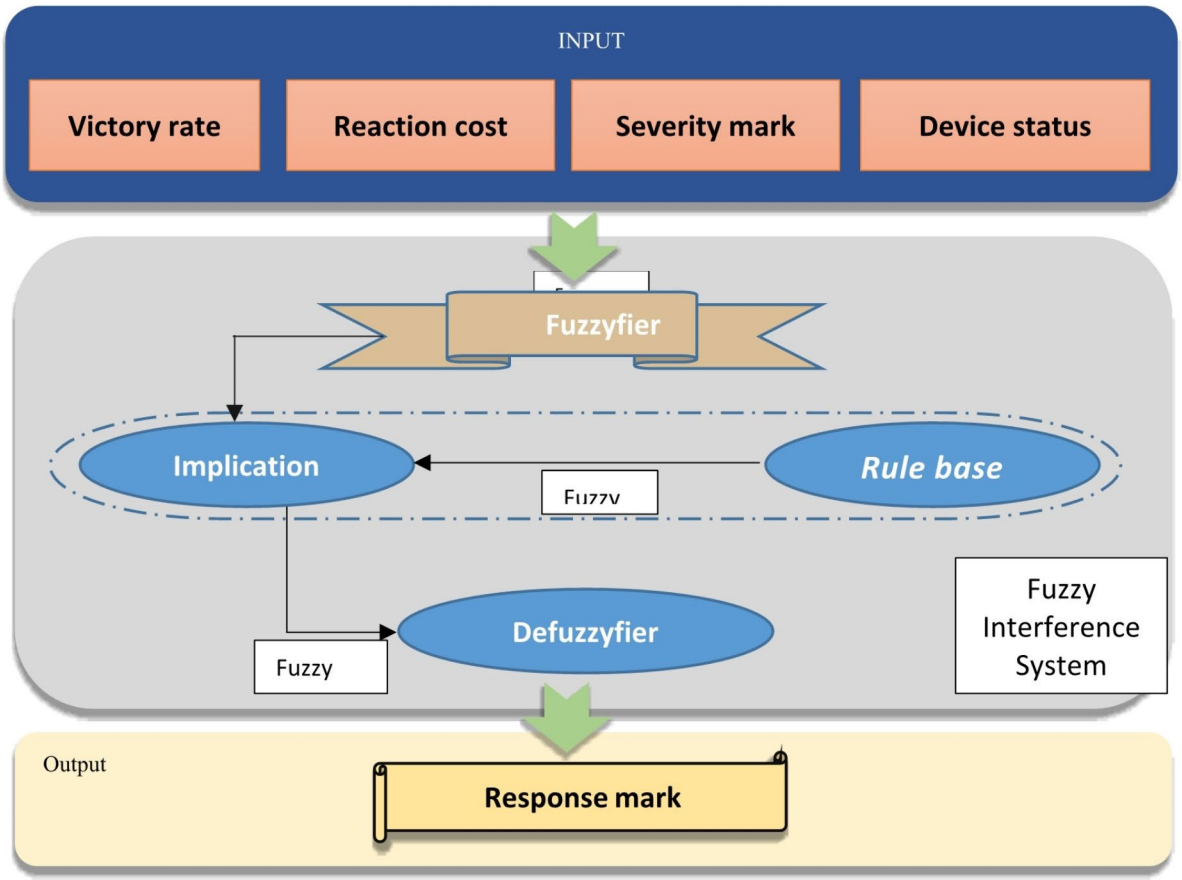


Fig. 3 Projected fuzzy inference scheme for reaction assortment

real-world IoT applications, ensuring they meet the necessary requirements for efficiency and effectiveness.

Determine the latency (L) between two points. Latency is the amount of time it takes for data to travel from its origin to its final destination.

$$L = \text{Ntend Tstart} \tag{2}$$

Which is the formula to use. Where L is the delay.

Data tends to arrive at its destination around the ten minute mark. At some point before Tstart, the data is sent. The total number of packets of data is N.

Bandwidth utilization (B) is the quantity of data that is sent across a network and may be calculated using this metric. Bandwidth utilization may be determined using the formula.

$$B = TD \tag{3}$$

Where D is the data rate in bits per second. Time spent transmitting data is denoted by T.

Third, the material Quality Index (CQI) is calculated, which provides a numerical representation of the quality of the given material.

$$CQI = N\,(1DmaxDi) \tag{4}$$

Where:

The content quality index abbreviated as "CQI". There will be N total frames. The i^{th} frame's distortion is denoted by Di. Moreover, Dmax is the maximum permissible distortion.

Estimating Energy Use (E)

Energy consumption is a quantitative measure of power usage over a certain time frame. The formula is.

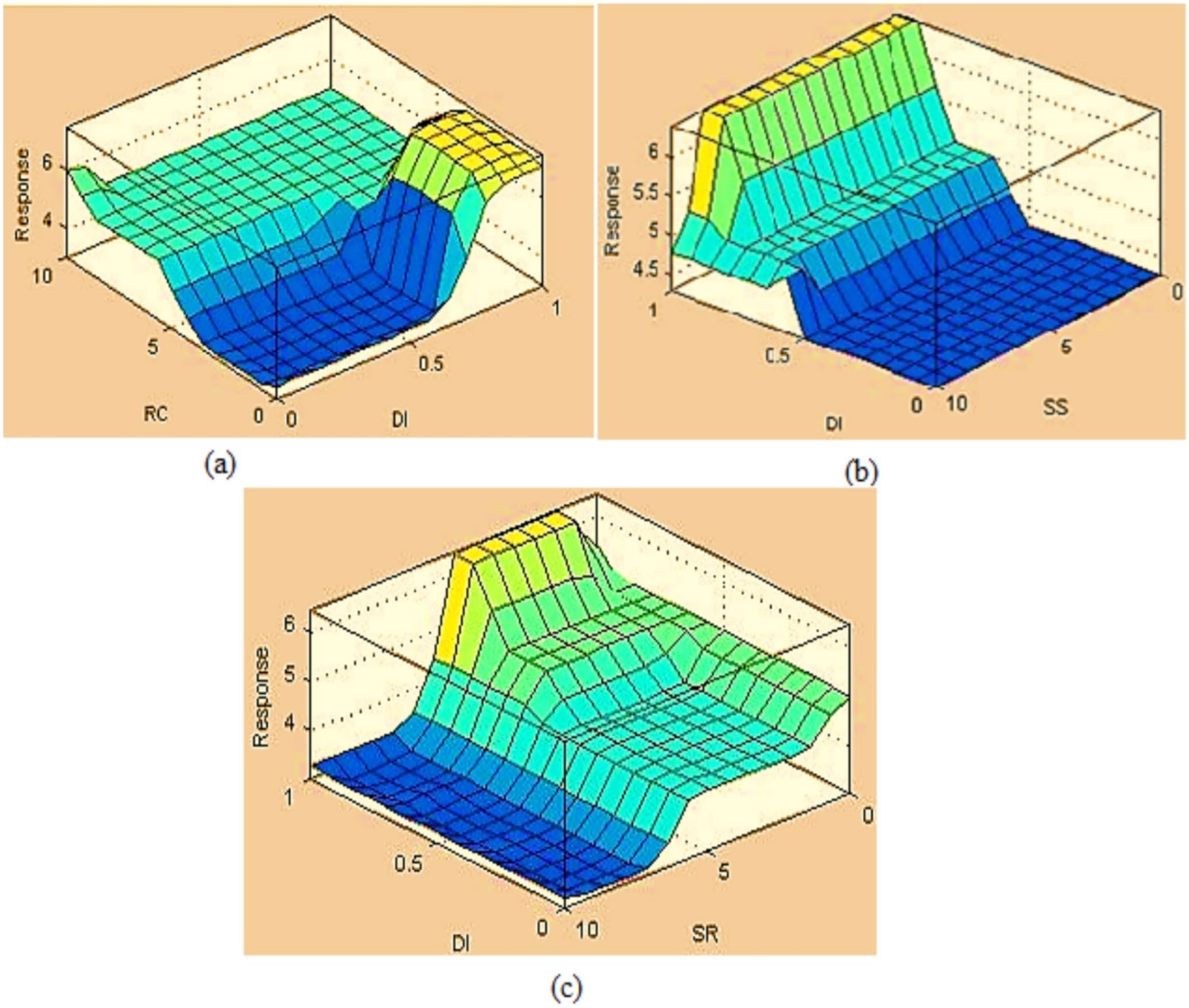


Fig. 4 Association between (**a**) Device Importance (DI), Response Cost (RC) and Response Score (RS) (**b**) DI, Severity Score (SS) and RS (**c**) DI, Success Rate (SR) and RS

$$E = P\,t - 4 \tag{5}$$

Where E is the amount of energy used.

The symbol "P" stands for "power.".

The period of energy consumption is denoted by t.

In order to characterise the IoT ecosystem and choose a suitable reaction to the assault scenario, four new types of metrics are offered. When making decisions about how to respond to an attack on the Internet of Things (IoT), fuzzy logic is employed to account for the lack of precision in the gathered data.

Figure 3 depicts the suggested method's fuzzy inference architecture. The membership functions for fuzzification, fuzzy rule, inference engine, and defuzzification are the major components in creating the fuzzy learning system.

Result and Discussion

Fuzzy logic is applied to response selection by using inference at edge nodes to interpret and act on the uncertain and imprecise data generated by IoT devices. This approach quantifies uncertainty in IoT data through fuzzy sets and rules, allowing the system to handle ambiguity effectively. Consequently, it enhances decision-making processes by providing more accurate and context-aware responses to potential security threats in IoT environments.

Table 2 Success rate computation for different measuring parameters in different implementation-based methods

Measuring Parameters		Implementation Methods	
		EDGE Based	Cloud Based
Severity Score	2	96.5	90.8
	4	96.7	91.2
	6	96.4	91.7
	8	97.2	90.7
	10	97.8	91.6
Device Importance	0.2	96.2	90.4
	0.4	96.8	91.2
	0.6	97.1	91.7
	0.8	96.3	90.8
	1.0	97.5	92.1
Response Cost	2	96.9	93.1
	4	97.3	92.9
	6	97.8	93.3
	8	97.2	92.2
	10	97.6	93.8

Response cost (RC), success rate (SR) and network load are used to model the proposed system's fuzzy inference engine. Here, an IoT network's wireless sensors undergo a simulated flooding attack to test out the suggested edge computing-based response selection. According to the experiments, the success rate and latency of the proposed response technique are both very high. An alarm is produced in a simulated flooding assault to prove the validity of the proposed strategy. Based on this notification, the suggested system chooses an action to take. The alarm is defined alongside the features of the associated IoT device and the task at hand. The technique is tested by altering alert parameters like those found in Internet of Things (IoT) devices and their uses.

Results from experiments on the proposed system are briefly addressed in terms of input-output correlation, fuzzy membership functions, response success rate, and latency measurements for cloud and edge-based implementations. Impact of assault in the IoT environment is employed as a qualitative and imprecise parameter for response selection in the suggested system. Therefore, fuzzy logic is employed to arrive at accurate conclusions from vague data. The decision is made based on qualitative data, but its efficacy is evaluated quantitatively through the rate of success of the chosen reaction action in the Internet of Things setting. Both reaction time and network load are used as quantitative measures of the proposed system's efficacy.

It is clear from Fig. 4 that there is a connection between the input and output variables. Figure 4 shows that an improvement in response score is associated with an increase in device importance and a decrease in response cost.

Success Rate (SR)

Each defined metric serves as a yardstick against which the success rate of applying the proposed fuzzy inference for response can be evaluated. The success rate of IoT projects does not fluctuate much as a result of environmental changes. The figure's most important conclusion is that the proposed fuzzy inference has a higher success rate at the edge level than at the cloud level. The closer-to-the-end edge nodes activate the response before the cloud nodes do. When using a cloud-based implementation, there is a significant lag time between detecting an attack and responding to it, which increases the intensity of the attack and complicates the attack scenario, rendering the chosen reaction ineffective. The success rate of responses increases because of the low delay in the edge-based implementation. The SR of r-response is found by using the formula.

$$SR = \left(\frac{s}{n}\right) X\ 100 \tag{6}$$

Where, s is the success counter's value, indicating how many times the desired action was taken, and n is the total number of times the desired action was chosen.

Device Importance

Device d is the victim, and the function to determine d's relative importance is.

$$DI = \frac{\sum\ (w\,(u)\,Xu)\,(w(\,v\,)Xv)\,(w(\,c\,)Xc)}{\sum\ w\,(u)\,w\,(v)\,w\,(c)} \tag{7}$$

where u is the device's data value, v is the importance of the function it performs, and c is the scalability with which its failure would be felt. On a scale from 0 to 1, DI is measured. A DI value of 0 denotes a somewhat unimportant item, while a value closer to 1 implies greater significance.

Response Score

Fuzzified inference engine output is defuzzified to produce clean inference results. The proposed method utilizes the centroid or center of gravity approach to defuzzification, one of many available approaches. The centroid method is used due to the convenience and reliability of its outcomes. Using the Equation below, we can calculate the final answer score using the centroid defuzzification approach.

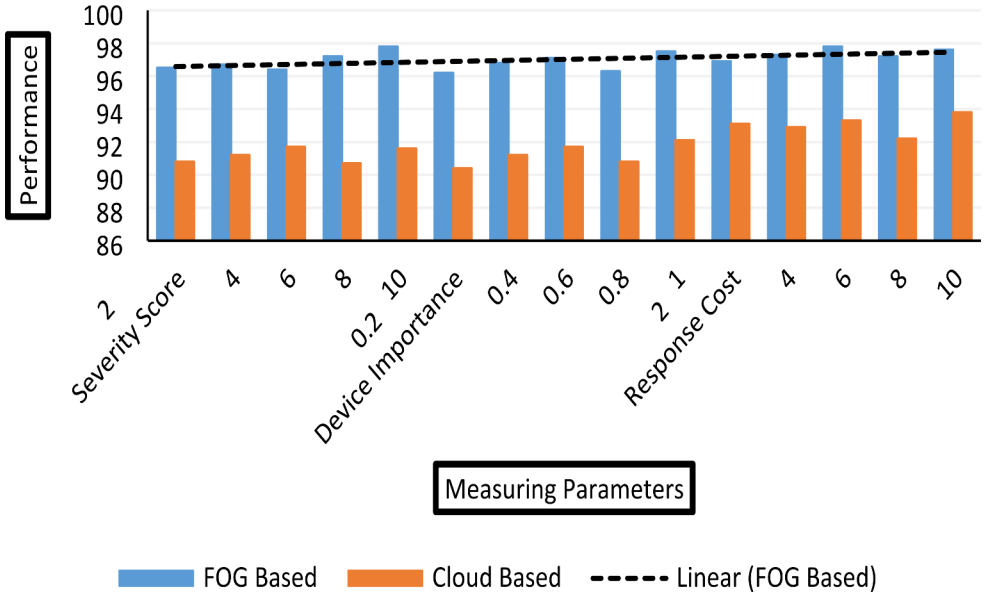


Fig. 5 Success rate computation for different measuring parameters in different implementation based methods

Table 3 Response time comparison for different implementation based methods

S. No.	Band width	Response Time (ms)	
		EDGE Based	Cloud Based
1	50	12.5	16.4
2	500	8.9	15.8
3	1000	6.2	14.7

$$RS = \frac{\int x\, r(x)}{\int r(x)} \tag{8}$$

Where, r(x) is the membership function and x is the corresponding fuzzy value.

As can be seen in Fig. 5; Table 2, the success rate does not significantly fluctuate in response to shifts in low, high, or medium metric values, i.e. shifts in the IoT environment. As can be seen in the figure, the proposed fuzzy inference has a far higher success rate in the edge than in the cloud, which is the main conclusion to be drawn from the data. The closer-to-the-end edge nodes activate the response before the cloud nodes do. When using a cloud-based implementation, there is a significant lag time between detecting an attack and responding to it, which increases the intensity of the attack and complicates the attack scenario, rendering the chosen reaction ineffective. When using an edge-based approach, there is less of a delay, which increases the response success rate.

Response Time

The proposed edge computing based self-defence method is also deployed on Azure as a centralised system for evaluation purposes. Security mechanisms in IoT applications should be evaluated based on how quickly they respond. Response C (RC) for Reaction R is defined as:

$$RC = \sum \frac{E + M + P}{300} X\ 100 \tag{9}$$

Where, E represents the amount of power, M represents the amount of memory, and P represents the amount of processing power needed to implement the solution. In a resource-limited IoT setting, responses with lower values of the response cost RC are preferred.

Since the impact of an attack might result in severe damage or loss of life in an IoT environment, the attack must be identified and neutralised at a faster pace to stop or minimise its impact. The suggested self-protection mechanism's response time for variable bandwidth in edge based implementation and cloud based implementation is depicted in Fig. 6 from Table 3. Due to the proximity of edge nodes to the end devices, reaction times in edge based implementation are roughly 26% lower compared to cloud based implementation. This allows for quicker assault detection and countermeasures.

Network Load Custom

The amount of data created by an IoT application grows proportionally with the number of connected devices, putting more stress on the underlying network. When opposed to the centralised cloud-based solution, edge-based computing significantly reduces network consumption since the load is distributed across a large number of edge nodes are summarized in Table 4. Figure 7 illustrates how the suggested approach, in comparison to

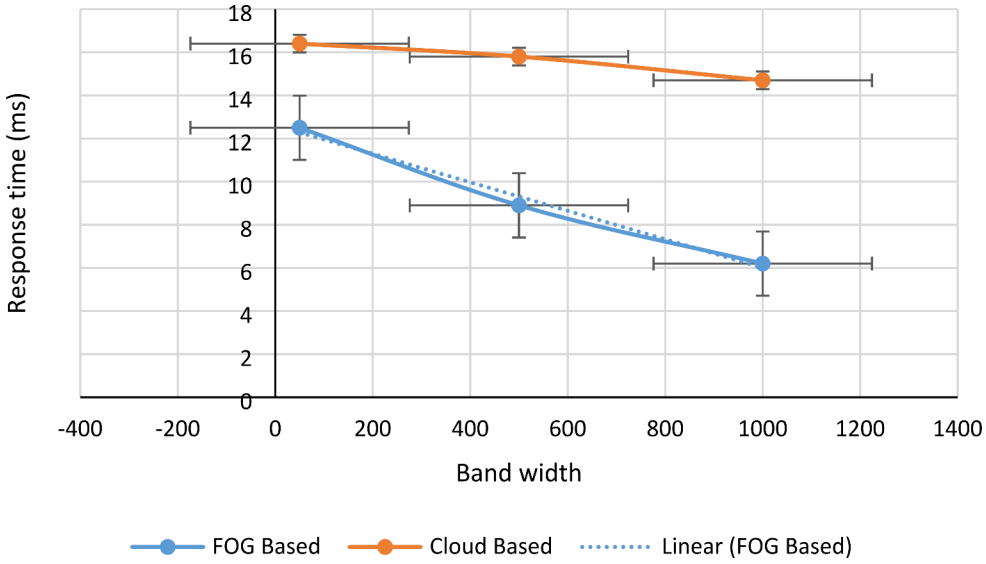


Fig. 6 Response time comparison for different implementation based methods

Table 4 Network load custom comparison for different implementation based methods

S. No.	Data Count	Network Load Custom (bits/sec.)	
		EDGE Based	Cloud Based
1	5000	0.32	0.46
2	10,000	0.71	1.12
3	15,000	1.09	1.15
4	20,000	1.32	2.16

cloud-based implementation, can dramatically lower network load as data volumes grow.

In general, the self-protection mechanism for the Internet of Things that is based on edge computing is able to detect and defend against cyber-attacks with reduced response time and network burden.

The diverse and heterogeneous nature of IoT devices, coupled with their constrained resources, renders traditional security measures ineffective. Conventional approaches struggle to address the unique vulnerabilities and communication protocols inherent to IoT ecosystems. This sets the stage for the proposed solution, which emphasizes a tailored, multi-layered security framework that leverages lightweight cryptographic algorithms, real-time threat detection, and adaptive security protocols specifically designed for IoT environments.

Conclusion and Future Work

The massive data flow produced by IoT-enabled devices must be intelligently interpreted by the security mechanism of IoT in order to detect both known threats and zero-day attacks. In this study, we suggest an edge node self-protection system for Internet of Things deployments. The suggested system can anticipate, identify, and

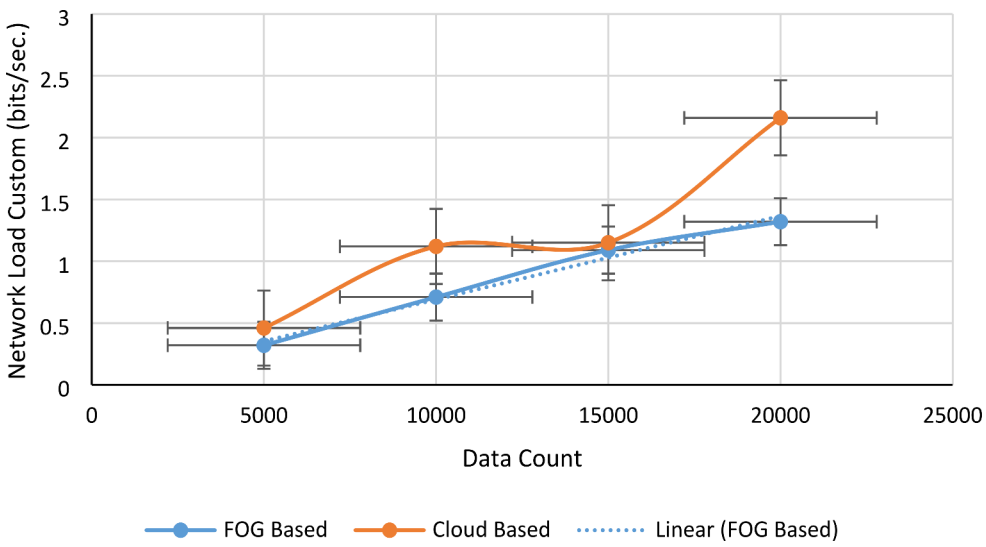


Fig. 7 Network load custom comparison for different implementation based methods

respond intelligently to cyber-attacks at a faster rate with less human interaction, all of which are crucial in an IoT setting.

Since IoT data is often imprecise and ambiguous, the suggested response selection mechanism employs fuzzy logic inference at edge nodes to make a decision. Few metrics are defined in the proposed system with the Internet of Things in mind. The defined metrics can be calculated rapidly at the edge nodes and include device importance, severity score, response cost, and success rate. In order to respond quickly to requests from end devices, the response model is deployed on distributed edge nodes. The proposed method has a nearly 7% greater response success rate in edge based implementation compared to the cloud, as shown by the simulation results. Therefore, the suggested model not only selects the proper response from the ambiguous IoT data, but also selects the response at a higher throughput. An evaluation of the suggested self-protection system demonstrates the efficacy of edge computing in terms of response time and network load.

Future work will focus on evaluating the sustainability of cerebral edge computing innovations for autonomously managing Internet of Things infrastructure. This includes assessing long-term efficiency, adaptability, and environmental impact to ensure robust and scalable IoT solutions.

Author Contributions Marcelo Zambrano-Vizuete: Consumption and design of study and Acquisition of the data, Juan Minango-Negrete: Analysis and interpretation of the data and Drafting, Wladimir Paredes-Parada: Formalization an editing, Review and investigation, Jorge Pérez-Chimborazo: Conceptualization, Ana Zambrano-Vizuete: Investigation and analysis.

Funding Not Applicable.

Data Availability Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

Conflict of Interest The authors declare no conflict of interest.

References

1. Balte A, Kashid A&, Patil B. Security issues in internet of things (IoT): a survey. Int J Adv Res Comput Sci Softw Eng. 2015;5(4):450–5.
2. Banerjee Anjishnu, David B, Dunson, Surya T, Tokdar. 'Efficient gaussian process regression for large datasets'. Biometrika. 2012;100(1):75–89.
3. Kolias C, Kambourakis G, Stavrou A&, Gritzalis S. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun Surv Tutorials. 2016;18(1):184–208.
4. Kott A, Wang C, Erbacher RF. Cyber defense and situational awareness'. Springer; 2015;62.
5. Roy V. An effective FOG computing based distributed forecasting of cyber-attacks in internet of things. J Cybersecur Inform Manage. 2023;12(2):8–17.
6. Riggins Frederick J. & Samuel Fosso Wamba 2015, 'Research directions on the adoption, usage, and impact of the internet of things through the use of big data analytics', 48th Hawaii International Conference on System Sciences (HICSS), pp. 1531–1540.
7. Dewanta F, Mambo MBPT, Scheme. Establishing trusted vehicular FOG computing service for rural area based on blockchain approach. IEEE Trans Veh Technol. 2021;70:1752–69.
8. Yakubu J, Abdulhamid SM, Christopher HA, Chiroma H, Abdullahi M. Security challenges in edge-computing environment: a systematic appraisal of current developments. J Reliab Intell Environ. 2019;5:209–33.
9. Roy V. A context-aware internet of things (IoT) founded approach to scheming an operative priority-based scheduling algorithms. J Cybersecur Inform Manage. 2024;13(1).
10. Wang J, Li D. Adaptive computing optimization in software-defined network-based industrial internet of things with FOG computing. Sensors. 2018;18:2509.
11. Sahil; Sood SK. FOG-cloud centric IoT-based cyber physical framework for panic oriented disaster evacuation in smart cities. Earth Sci Inf. 2022;15:1449–70.
12. Kumar P, Gupta GP, Tripathi RA, Distributed Ensemble. Design based intrusion detection system using FOG computing to protect the internet of things networks. J Ambient Intell Humaniz Comput. 2021;12:9555–72.
13. Saini K, Kalra S, Sood SK. Disaster emergency response framework for smart buildings. Future Gener Comput Syst. 2022;131:106–20.
14. Roy V, Roy L, Ahluwalia R, Khambra G, Ramesh M, Rajasekhar K. An advance implementation of machine learning techniques for the prediction of cervical cancer, 2023 IEEE International Conference on ICT in Business Industry & Government.
15. Villegas-Ch W, Barahona-Espinosa S, Gaibor-Naranjo W, Mera-Navarrete A. Model for the detection of falls with the use of artificial intelligence as an assistant for the care of the elderly. Computation. 2022;10:195.
16. Rezapour R, Asghari P, Javadi HHS, Ghanbari S. Security in FOG computing: a systematic review on issues, challenges and solutions. Comput Sci Rev. 2021;41:100421.
17. Wang S, Zhao T, Pang S. Task scheduling algorithm based on improved firework algorithm in FOG computing. IEEE Access. 2020;8:32385–94.
18. Ortiz-Garcés I, Villegas-Ch W. Model of telecommunications infrastructure for the deployment of technological services from the CLOUD to all the localities of the Ministry of Education in Ecuador. RISTI—Rev Iber Sist Tecnol Inf. 2019;E19:531–42.
19. Singh J, Singh P, Gill SSFOG, Computing. A taxonomy, systematic review, current trends and research challenges. J Parallel Distrib Comput. 2021;157:56–85.
20. Bellavista P, Berrocal J, Corradi A, Das SK, Foschini L, Zanni A. A survey on FOG computing for the internet of things. Pervasive Mob Comput. 2019;52:71–99.
21. Sarkar S, Chatterjee S, Misra S. Assessment of the suitability of FOG computing in the context of internet of things. IEEE Trans Cloud Comput. 2018;6:46–59.
22. Zhang PY, Zhou MC, Fortino G. Security and trust issues in FOG computing: a survey. Future Gener Comput Syst. 2018;88:16–27.

23. Hu P, Dhelim S, Ning H, Qiu T. Survey on FOG computing: architecture, key technologies, applications and open issues. J Netw Comput Appl. 2017;98:27–42.
24. Lera I, Guerrero C, Juiz CYAFS. A simulator for IoT scenarios in FOG computing. IEEE Access. 2019;7:91745–58.
25. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA. A comprehensive survey on FOG computing: state-of-the-art and research challenges. IEEE Commun Surv Tutor. 2018;20:416–64.
26. Moshayedi AJ, Roy AS, Taravet A, Liao L, Wu J, Gheisari MA. Secure traffic police remote sensing approach via a deep learning-based low-altitude vehicle speed detector through UAVs in smart cites: algorithm, implementation and evaluation. Future Transp. 2023;3:12.
27. Roy V. An improved image encryption consuming fusion transmutation and edge operator. J Cybersecur Inform Manage. 2021;8(1):42–52.
28. Tselios C, Politis I, Amaxilatis D, Akrivopoulos O, Chatzigiannakis I, Panagiotakis S, Markakis EK. Melding FOG computing and IoT for deploying secure, response-capable healthcare services in 5G and beyond. Sensors. 2022;22:3375.
29. Prabavathy S, Sundarakantham K, Mercy Shalinie S. Design of cognitive FOG computing for autonomic security system in critical infrastructure. J Univers Comput Sci. 2018;24(5):577–602.
30. Alrizq M, Stalin S, Alyami S, Roy V, Mishra A, Chandanan AK. Optimization of sensor node location utilizing artificial intelligence for mobile wireless sensor network. Wireless Netw. 2023:1–13.