mathematics

MDPI

Article

Hybrid Fuzzy Rule Algorithm and Trust Planning Mechanism for Robust Trust Management in IoT-Embedded Systems Integration

Nagireddy Venkata Rajasekhar Reddy [1], Pydimarri Padmaja [2], Miroslav Mahdal [3,*], Selvaraj Seerangan [4], Vrince Vimal [5], Vamsidhar Talasila [6] and Lenka Cepova [7]

[1] Department of IT, MLR Institute of Technology, Hyderabad 500043, India; drrajasekhar@mlrit.ac.in
[2] Department of Electronics and Communication Engineering, Teegala Krishna Reddy Engineering College, Hyderabad 500097, India; padmajavattem@gmail.com
[3] Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 17. Listopadu 2172/15, 70800 Ostrava, Czech Republic
[4] Department of Computer Science and Design, Kongu Engineering College, Perundurai, Erode 638060, India; selvaraj.cse@kongu.edu
[5] Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun 248002, India; vvi-mal@ec.iitr.ac.in
[6] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur 522302, India; talasila.vamsi@kluniversity.in
[7] Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 17. Listopadu 2172/15, 70800 Ostrava, Czech Republic; lenka.cepova@vsb.cz
* Correspondence: miroslav.mahdal@vsb.cz

Abstract: The Internet of Things (IoT) is rapidly expanding and becoming an integral part of daily life, increasing the potential for security threats such as malware or cyberattacks. Many embedded systems (ESs), responsible for handling sensitive data or facilitating secure online activities, must adhere to stringent security standards. For instance, payment processors employ security-critical components as distinct chips, maintaining physical separation from other network components to prevent the leakage of sensitive information such as cryptographic keys. Establishing a trusted environment in IoT and ESs, where interactions are based on the trust model of communication nodes, is a viable approach to enhance security in IoT and ESs. Although trust management (TM) has been extensively studied in distributed networks, IoT, and ESs, significant challenges remain for real-world implementation. In response, we propose a hybrid fuzzy rule algorithm (FRA) and trust planning mechanism (TPM), denoted FRA + TPM, for effective trust management and to bolster IoT and ESs reliability. The proposed system was evaluated against several conventional methods, yielding promising results: trust prediction accuracy (99%), energy consumption (53%), malicious node detection (98%), computation time (61 s), latency (1.7 ms), and throughput (9 Mbps).

Keywords: Internet of Things (IoT); embedded systems (ESs); trust management (TM); cyber-attacks; security

MSC: 94A16; 68T07; 68T20

Citation: Reddy, N.V.R.; Padmaja, P.; Mahdal, M.; Seerangan, S.; Vimal, V.; Talasila, V.; Cepova, L. Hybrid Fuzzy Rule Algorithm and Trust Planning Mechanism for Robust Trust Management in IoT-Embedded Systems Integration. *Mathematics* **2023**, *11*, 2546. https://doi.org/10.3390/math11112546

Academic Editor: Daniel-Ioan Curiac

Received: 2 May 2023
Revised: 24 May 2023
Accepted: 30 May 2023
Published: 1 June 2023

1. Introduction

The Internet of Things (IoT) is a new technology that has quickly acquired prominence. Smart buildings, smart hospitals, smart transit, environmental control, smart devices, online atmosphere, and other IoT applications have a significant impact on how people live their everyday lives [1]. IoT is a system in which networks of computer devices are linked to each other. Due to the diversity, resource limitations, and low computing energy of IoT devices,

there are several security and privacy concerns. These gadgets exchange data with minimal to no human contact [2]. Unfortunately, there are several security and trust issues brought on by IoT's growth. IoT devices are often found in distant areas, and the data they produce may be readily manipulated, resulting in false information, inconsistent data, and other protection and stability risks [3]. IoT safety has a big influence on how well IoT applications work. A guarantee of object reliability is required when an IoT object has to link with other items for information and communication safety purposes [4]. The study of organizational behavior offered by trust management is based on both current and past conduct. The adoption of trusted management may address issues with credential management, better user confidentiality, and data protection. There are now many suggested trust management solutions accessible on the IoT network [5]. While trying to evaluate the nodes' trust levels within an IoT network, many researchers significantly experience difficulties [6]. IoT devices are susceptible to security assaults because they operate in distant places in possibly hostile situations. These resource-constrained systems, unfortunately, are unable to implement standard safety techniques since they need strong hardware and software [7]. One may assume that a sizable quantity of critical information is handled in IoT devices given the size of IoT and the sectors that use this innovation. As a result, safety is essential. Trust management is one technique utilized to evaluate the network's dependability [8]. By providing a trust value to each node and identifying its degree of trust, trust management tries to assure the dependability of the network. As a result, the data supplied by a node with a higher trust value are regarded as accurate. The trustor and the trustee are the two parties who should be engaged to establish a trust contract [9].

Several sophisticated technologies, such as portable industrial robotics and automobile technologies, are driven by contemporary ESs. The ES is vulnerable to disruptions or abnormalities, particularly in such natural situations. The effects change with time while functioning in a non-stationary setting. For instance, the reliability of a sensor relies on its environment. Moreover, interior states such as the assessment of a mobile robot's present location may become more unreliable and unpredictable. As a result, the system's functioning becomes riskier. Conversely, safe functioning is essential, particularly with human involvement. Hence, as requirements for embedded systems rise, so do the chances of errors and unusual behavior. The former is mostly caused by the growing volume of and dependence on sensory input, as well as by being immersed in an atmosphere that is becoming more complicated and even disruptive and unpredictable [10]. In addition to these hardware difficulties, the programs utilized within the data flow can also induce difficulty due to inherent ambiguity in their outputs. Much worse, these ambiguities change during the system's lifespan, for example, as a result of drifting and fading impacts. These inconsistencies make the fundamental knowledge less trustworthy and add ambiguity and confusion, which seriously impairs safe functioning. Since the activities of an embedded system are dictated by the sensed data and analysis, which in turn affect the general security of the device and its connection with the surroundings, trust or trustworthiness is the key idea here. Hence, appropriate methods must be developed to handle the ambiguities or reliability, as appropriate [11]. The new area of data collection and transmission without human involvement comprises IoT and ES. It is described as a network of interconnected items that have embedded sensors, computers, and control mechanisms. IoT technology evolved as a consequence of scientific advancements. Applications for IoT and ESs are becoming more prominent in practically every industry. IoT applications integrate smarter cities and households, improve transportation and logistics, support healthcare applications, and monitor the environment and public safety. Industrial robots, the aerospace industry, networks and communications, and the automobile industry are a few examples of embedded system applications. Figure 1 displays a few of the predominant IoT and ESs applications [12].

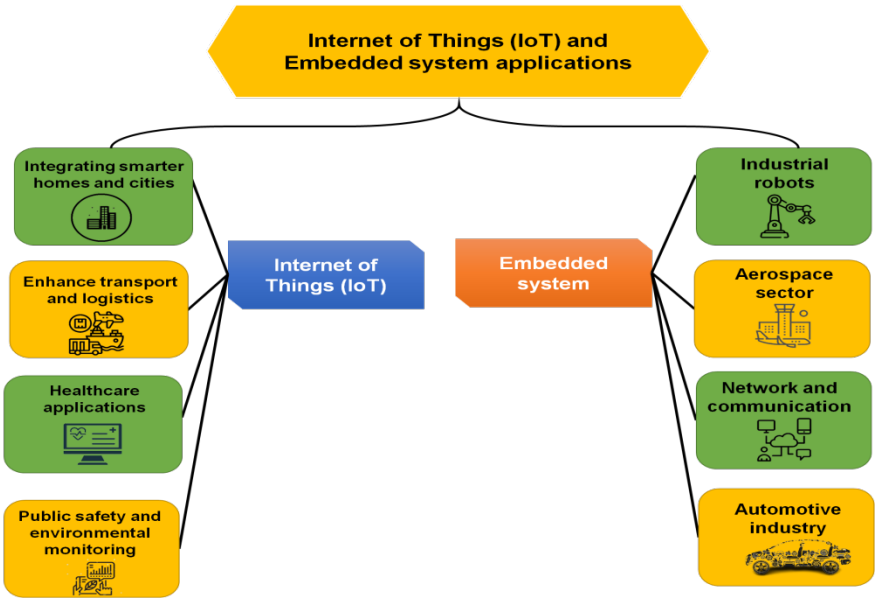

Figure 1. Applications of IoT and ES.

The IoT and ESs allow a variety of applications, yet they also present some security concerns. The idea of a conceptual approach for so-called trust management, which consists of infrastructure and certain procedures, will be proposed as a resolution. Its goal is to boost user confidence in framework efficiency to strengthen the framework's resilience and, as a result, enable efficiency optimization without compromising security.

The remaining parts are organized as follows: The problem statement and literature review are covered in Part II, followed by the suggested methodology in Part III, the proposed methodology's assessment in Part IV, and the conclusion in Part V.

2. Literature Survey

The suggested method was planned after referring to the relevant research works of various researchers on the aspects of Internet of Things, embedded systems, fuzzy rule algorithms, and trust planning mechanisms.

2.1. Review Based on Internet of Things

One of the key ideas in IoT is wireless sensor platforms, which include many network devices with limited resources. These asset-limited sensor nodes now face increased security risks as a result of IoT's widespread adoption. Praveena et al. [13] includes a comprehensive overview of the IoT, discussing its key features, enabling technologies, and potential applications in various domains such as healthcare, transportation, and smart cities. They highlighted the need for efficient communication protocols, scalable architectures, and security mechanisms to ensure the successful deployment and operation of IoT systems and suggested a blockchain-based trust management system (BBTM) to address the issue, wherein portable interface nodes would assess the reliability of IoT devices. For trust assessment, BBTM may design decentralized applications and validate the calculation procedure. The capacity and skill to recognize trustworthiness need more time. Trust is important in providing consumers of the IoT system with an efficient, dependable, expandable, and trusted setting where they may exchange personal data on a secured interaction network. Consequently, TM is a key component of a large-scale IoT platform's ability to transmit data successfully and securely across different nodes. For commercial IoT nodes that are unable to secure information, a lightweight strategy is suggested to address security concerns, delay, and the danger of malevolent actions [14].

2.2. Review Based on Embedded Systems

The evolution of and advancements in embedded systems have led to the proliferation of many embedded systems in various domains. Bitencourt et al. [15], combining embeddings and fuzzy time series for high-dimensional time series forecasting in (i), presented a comprehensive survey on many embedded systems, discussing their characteristics, challenges, and applications. They highlighted the key design considerations, including energy efficiency, real-time constraints, and communication protocols, to achieve many reliable and scalable embedded systems. In both controlled and distributed designs, the effect of rogue gadgets has been reduced through TM. Unfortunately, the majority of these conventional TM frameworks have difficulties with computing, memory, and transmission. The substantial number of factors that need to be modified is this system's primary shortcoming. For massive IoT operations, transferring reliable information across trusted parties is essential. However, the absence of trustworthy partnerships across IoT firms creates considerable obstacles to the aforementioned aim. The authors of [16] suggested a lightweight, secure, triple-trusting architecture (SLTA) that effectively utilizes a supportive system for blockchains. The design features a decentralized access control system that improves privacy and authority over digital records as well as an oracle-based information-collecting system that assures that the information gathered from IoT edge equipment is not altered. The power consumption for the computation is very high. Investigators require IoT-specific technologies, methodologies, and information to increase the degree of privacy for IoT.

2.3. Review Based on Fuzzy Rule Algorithms

Fahim et al. [17] offered an architecture for creating IoT context-aware safety mechanisms to identify malicious information in IoT application situations. To enhance the performance and efficiency of fuzzy rule algorithms, researchers have proposed various extensions and modifications. They introduced the concept of adaptive neuro-fuzzy inference systems (ANFIS), which combine fuzzy logic with neural networks to improve learning and adaptability. ANFIS has been applied in various applications, including system identification, time series prediction, and fault diagnosis. IoT-Flock is a recently developed, open source IoT information-generating program that makes the foundation. Investigators may create an IoT usage instance with both legitimate and malicious IoT systems using the IoT-Flock tool and create data. The suggested architecture also includes an open access application for transforming the traffic that IoT-Flock collects into an IoT database. The reliability of the Internet of Things (IoT) system depends on the detection of anomalous and malicious information. Diwan et al. [18] introduced a simple, limited-cost, greater privacy; lightweight, low-computational-time feature selection IoT intrusion detection approach feature entropy estimation (FEE). The link between all retrieved characteristics was then validated using FEE to detect fraudulent traffic in IoT systems. Due to the implementation of several commands, it is quite complicated. One of the major problems with the Android environment's cyber protection is the proliferation of malignant applications. It is almost impossible to manually identify malware applications in the Android environment due to the rapid growth of malware deployments for the platform. Machine learning is already a young method for detecting malware as a consequence. Abawajy et al. [19] analyzed how frequently utilized filter-based feature-selecting approaches function, with a focus on Android malware detection, and formulated the feature-selecting issue as a quadratic computing issue. The needed training time was longer [20,21]. The malicious program posed a danger to the safety of organizations and networking. Obfuscated malware that may avoid investigation and recognition by anti-malware programs has grown common, although anti-malware solutions are meant to safeguard devices and connections from malware assaults. As a result, a key worry now is how to identify and eliminate malware that has been hidden from the networks.

2.4. Reviews Based on Trust Planning Mechanism

In the context of IoT, trust planning mechanisms are crucial for ensuring the reliability and security of interconnected devices and systems. Mazumdar et al. [11] proposed a trust evaluation model for IoT environments based on fuzzy logic and multidimensional trust factors. They considered factors such as device reliability, communication quality, and security behavior to assess the trustworthiness of IoT devices, contributing to the effective management of trust in IoT ecosystems. In this study, a semi-supervised solution for obfuscated malware identification was proposed that combines deep learning, extraction of features, imagery modification, and computing approaches. With the advent of 5G and above systems, data protection and security have become key problems, giving rise to innovations such as blockchain and federated learning. They enable autonomous data to be used to train machine learning algorithms while maintaining anonymity. This research [22] addressed security concerns unique to this new framework for learning and explored the prospects for IoT malware identification made possible by federated learning (FL). In this regard, a methodology for detecting malware that affects IoT systems utilizing FL was provided. Because of the broad variety of sensitive apps that the Android system has embraced, including banking services, it is increasingly being targeted by malware that takes advantage of security system weaknesses. Methods for the identification of mobile malware have been suggested in a few studies. To obtain the highest level of effectiveness and performance, however, changes are necessary. To identify suspicious Android-directed assaults, we applied machine learning and deep learning technologies. The computation cost is very high [23]. Conventional network equipment has long encountered difficulties with security and reputation administration. Hence, this study provides software-defined networking (SDN), a unique scalable system for credential and trust management of IoT systems in IoT connectivity, a satisfactory proof of concept that shows the sustainability of the suggested system, using modeling that can hold the shared key of IoT systems on the blockchain and properly route networked congestion using SDN.

3. Problem Statement

Over the past several years, millions of IoT and ESs systems without adequate security features have been produced and deployed, and this number will continue to grow as breakthrough innovations become available. Notwithstanding the above information, IoT and ES's security problems are widespread. By taking advantage of these privacy holes, attackers may create a false network and carry out local or distant operations. Moreover, they can alter confidential data without authorization, stop the IoT from functioning normally, or even completely harm the IoT technology. IoT and ES's hardware and software elements are both potentially vulnerable. The demand for effective approaches to detect attacks in IoT and ESs systems within networks is driven by their susceptibility to assaults. To solve these security concerns, we thus introduced the TM-based FRA + TPM technique.

4. Research Methodology

IoT and ESs are effective technologies for many smart applications and have become a potent medium for sharing data and expressing/debating ideas. High-quality safety is a constant need to protect the sensitive data sent between ESs and IoT. One of the main issues to be addressed is safe routing to stop network impersonating attacks. Thus, we recommended the fuzzy rule algorithm and trust planning mechanism (FRA + TPM) for efficient trust management and to enhance IoT and ESs security.

4.1. Fuzzy Rule Algorithm (FRA)

We created FRA, an efficient trust management system for IoT and smarter environments. The FRA defines the mechanism's functioning and it explains how the proposed technique would operate. The IoT system enables the devices to join and terminates the connection at any time. IoT is a very volatile network as a result of this feature. The goal is to provide a technique that collects the network's adaptive properties to enhance the IoT

platforms' overall architecture. Figure 2 shows the FRA procedure, where a server assesses the querying node to which it is sending the resources. When there are plenty of querying nodes, the FRA process may employ the First in First out (FIFO) activity to rank them in order of importance. After choosing a node and providing capabilities to it, the server changes its local storage to the transactions' results. The FRA is employed to assess trust in the IoT network ecosystem in the framework of trust in traditional cultures. We focused on the review process rather than the prioritization strategy for choosing the querying node.

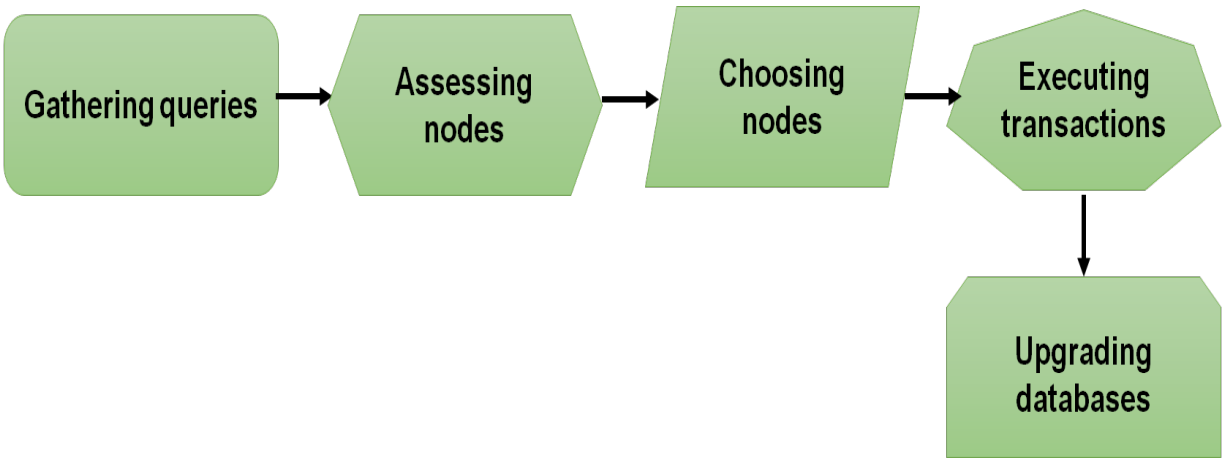

Figure 2. Process of FRA method.

Establishing and preserving trust between the IoT nodes of the intelligent system is the FRA's main goal. The FRA applies fuzzy logic computing to determine the reliability of IoT nodes. The server assesses each node's reliability after receiving a query from it before assigning a specific function to it. Utilizing trust qualities, such as each node's authenticity, suitability, and sensitivity to other nodes in the system, trust is assessed. When a node sends a demand to the server for any kind of operation, the system paradigm begins to function. The resource supplier (server) initially assesses the authorizing node using predefined trust characteristics, and only offers capabilities to trustworthy nodes if necessary. If not, the server considers the receiving node to be malignant or untrustworthy. The trust level is calculated utilizing fuzzy logic computing, where fuzzy rules are implemented to each querying node to maintain trust between the server and the querying node. Furthermore, the server receives the finalized trust value of the querying node after fuzzy logic analysis based on preset trust characteristics. The server chooses whether to serve the node or ignore it based on the ultimate trust value. To remove malignant nodes, a cutoff (CF) level is also specified. The FRA introduces a fuzzy filter that comprises two processes with predetermined thresholds.

The resulting trust level is re-evaluated by the fuzzy filter and compared to the predetermined cutoff value. The CF value could be flexible, changing in response to service requests. The threshold level is larger for extremely reactive equipment, while it is lower for less susceptible ones. The server offers resources to a node when its score is higher than or equal to the preset cutoff level, and once the operation is complete, the server changes its storage appropriately. The server refuses to send resources to the querying node and labels it as malignant or untrustworthy if the node does not reach the minimum cutoff value. The time-driven technique is utilized to increase the IoT platform's degree of confidence. A particular timestamp is assigned to the querying node, and the trust level of the querying node is saved in the storage at the moment the querying node is assessed by the server. There is no requirement to re-evaluate the node if it demands the resources a second time during the given period; instead, the server utilizes the node's previous trust level. The server must re-evaluate the querying node's trust level if the node demands resources after the specified time has passed. The period is fixed to 60 s for each node assessment. Moreover, every node is re-evaluated when the time has passed,

and the storage is upgraded with the properly measured trust level for that specific node. Fuzzy filters are crucial for removing fraudulent nodes because they utilize methods to contrast calculated trust with a cutoff level. The server defines the asking node as trusted if the calculated trust is equivalent to or higher than the cutoff level; otherwise, the server identifies the querying node as malignant. The calculated trust of a node that the server flags as malignant is not kept in the server's internal storage. We developed an algorithm called Assessing Trustworthiness that analyzes the trustworthiness of arriving nodes in the system utilizing a predetermined cutoff level. The method in Algorithm 1 describes how the network operates for reliable nodes.

Algorithm 1: Trust node prediction

Configuration
m_j-one Node for each query
FC: Collection of fuzzy scores
F = First trust score
$L_T^{(T)}$ = Last trust score
$P_C^{(T)}$ = Prior determined cutoff score
n_T = Trusted node
U : $(0.04 \rightarrow high(+), low(-))$
Outcome: Trusted Node
Process
Start
$Q_i^{(New)} \rightarrow U$
$n_i^q \rightarrow U$
If U receive $\left(q, n_i, M_T^T\right)$ then
$N_i^{(R)} \rightarrow n_i(N)$
If $\left(N_i^{(R)} \geq P_T\right) then$
$L_T^{(T)} \leftarrow P_T + U$
Else if $\left(N_i^{(R)} \geq P_T\right) then$
$L_T^{(T)} \leftarrow P_T - U$
End if
End if
$L_T^{(T)} \rightarrow$ Collection
If $\left(L_T^{(T)} \geq H_T^{(T)}\right)$ then
$M_{n_{(i)}}^{(T)} \leftarrow n_i$
$M_n^{(T)} \leftarrow$ Solutions
$D_S \leftarrow M_{n_{(i)}}^{(T)}\left(L_T^{(T)}\right)$
else
$U_n \leftarrow n_i$
$m_j \rightarrow V_t$
End if
End

Description of the symbols used in Algorithm 1:
Symbols in Configuration:
m_j: One node for each query.
FC: The collection of fuzzy scores.
F: The first trust score.
$L_T^{(T)}$: The last trust score.
$P_C^{(T)}$: The prior determined cutoff score.
n_T: The trusted node.
U: (0.04→high (+), low (−)): A membership function or fuzzy set representing the uncertainty level. Here, the value U ranges from 0.04 to high (+) or low (−).

Outcome: The trusted node as the outcome of the process.

Symbols in Process:

$Q_i^{(New)}$: The new query input.

n_i^q: A specific node associated with the query.

U receive $\left(q, n_i, M_T^T\right)$: The condition where U receives a query, associated node, and a trust score.

$N_i^{(R)}$: The trust score received from node N.

P_T: A predetermined threshold or cutoff score.

$L_T^{(T)} \leftarrow P_T + U$: The update of the last trust score with the addition of U.

$L_T^{(T)} \leftarrow P_T - U$: The update of the last trust score with the subtraction of U.

$H_T^{(T)}$: A high trust score threshold.

$M_{n_{(i)}}^{(T)}$: The trusted node associated with query n_i.

$M_n^{(T)}$: The solutions or outcome based on the trusted node.

D_S: A collection of solutions associated with the trusted node $M_{n_{(i)}}^{(T)}\left(L_T^{(T)}\right)$ and the last trust score $L_T^{(T)}$.

U_n: A specific node denoted as uncertain.

$m_j \rightarrow V_t$: The transfer or assignment of query node m_j to a variable V_t.

This algorithm, Identifying Malicious Node, which functions as a filtration, was created for malignant node identification. As soon as a node has finished using the reasoning engine and the server has received the final trust level from the querying node, this method is once again evaluated on that specific node. This method labels a node as malignant or untrusted if it has lower trust than the cutoff level. The FRA is effective and clever in identifying malevolent or hacked nodes and taking appropriate action. Two trust properties, namely, a node's reliability and flexibility, are specified to calculate a node's trust level. Reliability, a trust characteristic, indicates the degree of certainty and trust for a querying node. Based on its scope, we split the attribute reliability into three groups, including (0, 1). Flexibility, a trust characteristic, denotes a node's interoperability with the server and if a node may legitimately demonstrate its trust in the trustworthiness node. The characteristic flexibility is also separated into three groups based on input data. We obtain a clear range of outcomes for the property total trust by employing these input data to the inference engine along with the fuzzy system. The detail of detecting malicious nodes is provided in Algorithm 2. The suggested mechanism's entire workflow is shown in Figure 3.

Description of the symbols used in Algorithm 2:

Symbols in Configuration:

$M_{n_{(i)}}^{(T)}$: The last trust value associated with node n_i.

FC: The collection of fuzzy scores.

$P_C^{(T)}$: The prior determined cutoff score.

T_n: The novel trust score used for malignant node detection.

U: Trust value up/down (0.05) per request: A membership function or fuzzy set representing the trust value adjustment per request. Here, the value U is 0.05, which can be added or subtracted to the trust score.

Outcome: The malignant node as the outcome of the procedure.

Symbols in Procedure:

$A_n^{(T)}$: The trust value associated with node n_i.

$U_{receive} A_n^{(T)}$: The condition where U receives the trust value $A_n^{(T)}$.

$T_{Pr}^{(T)}$: A temporary trust score.

$P_{req}^{(T)}$: A requested trust score.

T_n: The updated trust score for the node.

$T_n \rightarrow S$: The transfer or assignment of the trust score T_n to a variable S.

$P_C^{(T)}$: The cutoff score or threshold.

Malicious: A malignant node.
$n_i^{(B)}$: A specific node categorized as a malignant node.
Allowed: A node that is allowed or trusted.
$n_i^{(R)}$: A specific node categorized as a trusted node.

Algorithm 2: Malignant nodes detection

Configuration
$M_{n_{(i)}}^{(T)} =$ last trust value of n_i
FC = Collection of fuzzy scores
$P_C^{(T)}$ = Prior determined cutoff score
T_n= Novel trust score for malignant node detection
U: Trust value up/down (0.05) per request
Outcome: Malignant node
Procedure:
Initialize
$A_n^{(T)} \leftarrow M_{n_{(i)}}^{(T)}$
$if\ U_{receive} A_n^{(T)} \leftarrow M_{n_{(i)}}^{(T)}$ then
$T_{Pr}^{(T)} \leftarrow A_n^{(T)}$
$if\left(P_{req}^{(T)} \geq P_C^{(T)}\right) then$
$T_n \leftarrow A_n^{(T)} + U$
$else\ if\left(T_{Pr}^{(T)} \geq P_C^{(T)}\right)$ then
$T_n \leftarrow A_n^{(T)} - U$
$end\ if$
$end\ if$
$T_n \rightarrow S$
$if\ T_n \leftarrow P_C^{(T)}$ then
$Malicious \leftarrow n_i^{(B)}$
Else
$Allowed \leftarrow n_i^{(R)}$
$end\ if$
End

4.2. Trust Planning Mechanism (TPM)

The TPM is intended for significant platforms with a collection of predetermined functions as shown in Figure 4. It was designed specifically for embedded systems utilized in mission-critical, safety, and security-related activities. A dependable operator initializes and configures the system, and it is not anticipated that the user would make changes to it while using it. Basic embedded system elements such as the CPU, RAM, system bus, a peripheral bus, peripherals, and trusted planning additions such as the trusted zone, planner, application-aware MPU, time-slice and synchronization analyzers, and a peripheral bus administrator are all included in TPM. The planner controls how programs are performed and notifies the trusted zone when a novel service has to be run. Programs are started by the planner either by its predetermined plan or in reaction to outside circumstances. As a result, the planner is set up with the planning rules it has to follow and is notified of the active processes on the platform. The trusted zone preserves the program's present condition and passes a command to the novel program when a novel program is planned to run. The trusted zone deploys and enables the appropriate CPU time-slice tracker (for example, a timing interruption) and the synchronization analyzers that transmit responsibility back to it from the program before the program is executed. The trusted zone may always reclaim management of the CPU with the least amount of delay thanks to the CPU duration and synchronization analyzers working in combination. As the planner operates effectively on the CPU and is separated from the programs and various platform parts, it is constantly able to pause and resume the operation of programs.

Every program executing on the platform is assigned a unique specialized memory sector, and an assessment knowing storage security module (SSM) enforces application borders to guarantee activity separation. Lastly, we implement a peripheral bus administrator to guarantee that bus connection to the peripherals is safely managed. The CPU (the executing apps) and other good peripherals cannot be denied connection to the peripheral bus by misbehaving devices thanks to the peripheral bus controller, which manages access from the different peripherals. Moreover, it makes sure that every program is limited to using the peripherals permitted in the trusted zone.

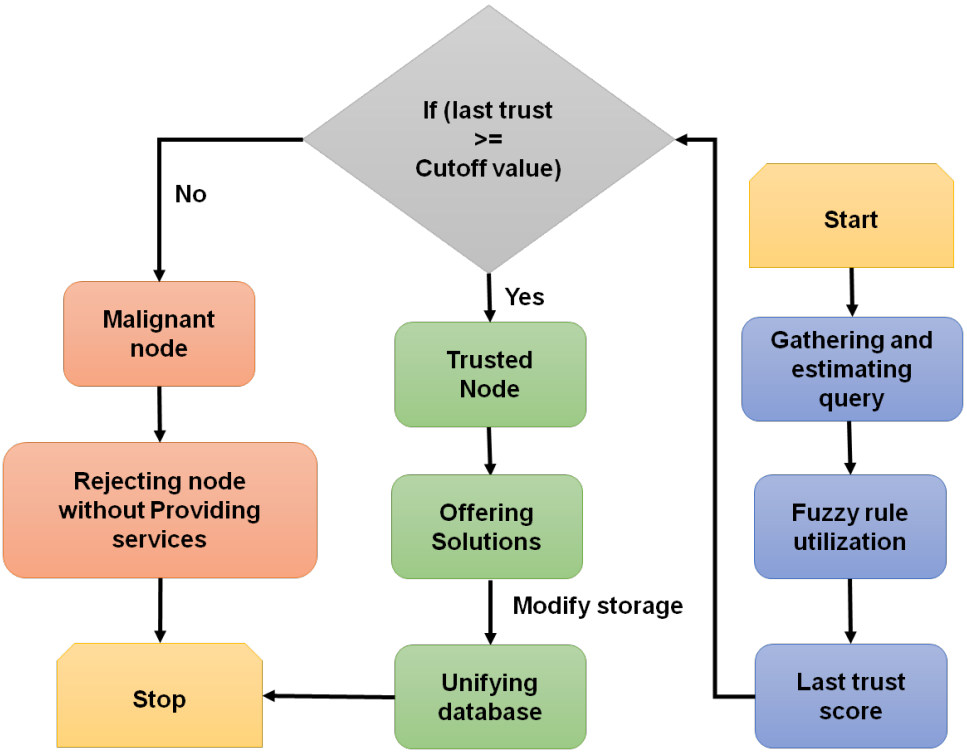

Figure 3. Workflow of proposed FRA.

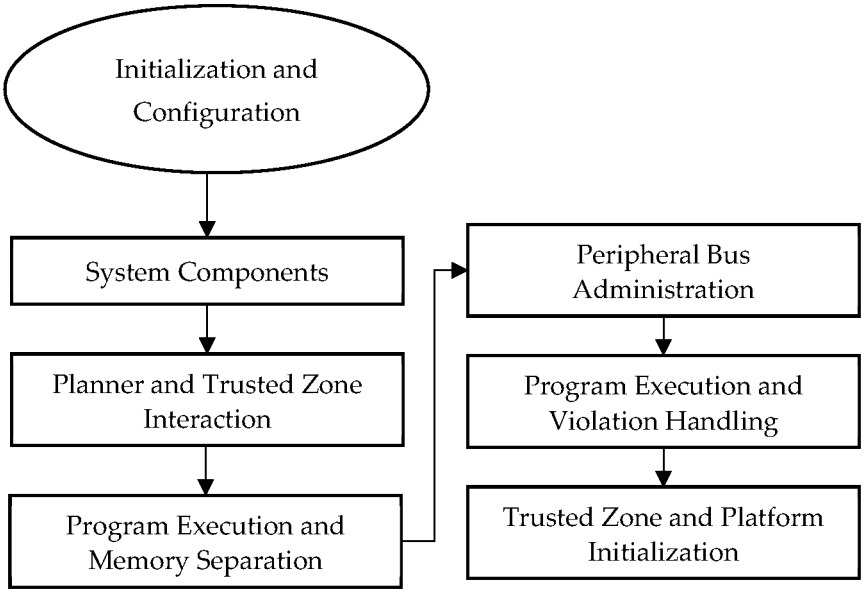

Figure 4. Flow of TM module.

Every program begins running at the first command it receives and keeps running until it crashes, breaks a security rule, or is interrupted. Whenever a program attempts to

access or alter a program or content that does not correspond to it, a privacy violation is triggered. Moreover, if a program attempts to perform an atomic segment that exceeds the predetermined limit or if its allotted CPU period runs out, a safety violation is generated. Moreover, if any program attempts to use a service to which it has not been allowed access or if the duration of any bus operation is beyond a pre-maximum limit, the peripheral bus administrator generates a safety violation. As an alternative, the atomicity monitor may also be used to impose the maximum limit on the duration of bus operations. Responsibility is returned to the trusted zone (through the hardware planner) when an activity is interrupted or forcefully ended as a result of a safety violation. The device planner then continues the operation of any outstanding programs.

We presume that a trustworthy manager provides platform initiation parameters to the trusted zone in terms of platform activation. When the device is turned on, the trusted zone runs first and initiates system startup using the executive's commands. The number of apps, the peripherals they utilize, the storage architecture, the operation plan, and preemption rules (such as the maximal CPU time-slice for every program, etc.) are a few examples of these factors, although they are not restricted to them. The scheduler is set up by the trusted zone to run programs regularly or in response to events. Moreover, it sets up specific software and data divisions for every activity by configuring the asset allocator's CPU scheduling settings and the software SSM. The trusted zone also constructs the peripheral bus administrator with knowledge of the services' needs for peripheral connectivity after initializing the software storage of each unique activity.

5. Result and Discussion

In this part, the computation outcomes of the suggested scheme are compared to those of the conventional strategies in terms of trust level prediction, energy consumption, malignant node detection, computation time, latency, and throughput. The existing comparison systems include Trust2Vec, Trust-FTSR, RFA, and EMBTR.

5.1. Trust Prediction Accuracy (%)

Predicting the amount of trust among organizations and consumers who are not already linked is the procedure of forecasting a new trustworthy relationship. Clients must be able to estimate the trust level to have trustworthy communication in IoT and ESs applications. The trust prediction accuracy of the suggested and conventional systems is shown in Figure 5. The outcome of the trust prediction accuracy is depicted in Table 1. This demonstrates that the proposed system's indicated trust level forecast is accurate.

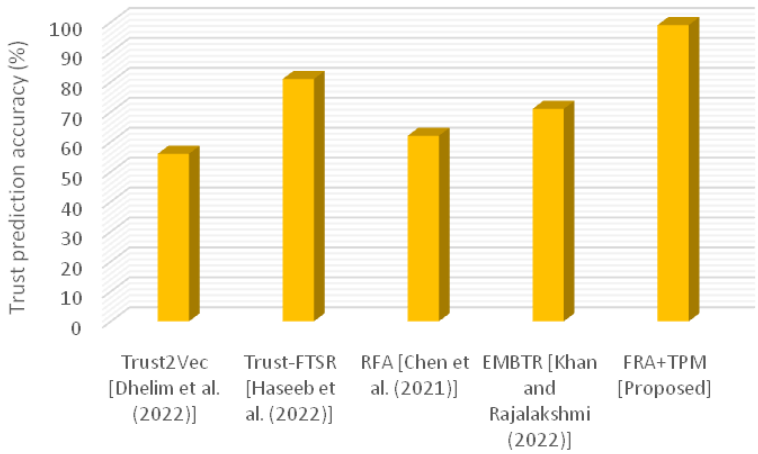

Figure 5. Trust prediction accuracy for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 1. Outcome of trust prediction accuracy.

Methods	Trust Prediction Accuracy (%)
Trust2Vec [24]	56
Trust-FTSR [25]	81
RFA [26]	62
EMBTR [27]	71
FRA + TPM [Proposed]	99

5.2. Throughput (Mbps)

The quantity of data efficiently sent between two locations in a certain length of the period is known as throughput. It may also be defined as the highest required capability that a method can handle. The throughput results using the suggested and conventional systems are shown in Figure 6. The outcome of the throughput is depicted in Table 2. It demonstrates that the recommended approach offers more throughput than the existing strategies.

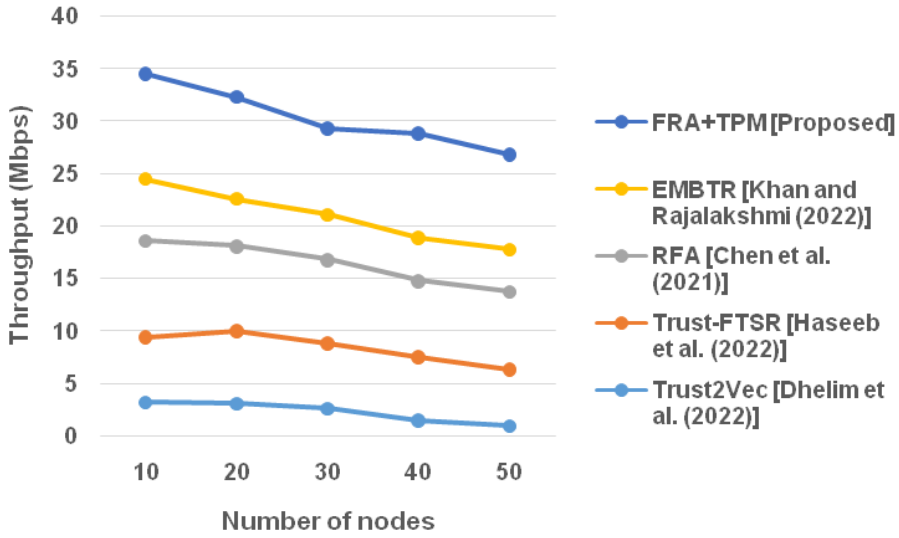

Figure 6. Throughput for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 2. Outcome of throughput.

Number of Nodes	Throughput (Mbps)				
	Trust2Vec [24]	Trust-FTSR [25]	RFA [26]	EMBTR [27]	FRA + TPM [Proposed]
10	3.2	6.2	9.2	5.9	10
20	3.1	6.9	8.1	4.5	9.7
30	2.6	6.2	8	4.3	8.2
40	1.5	6	7.3	4.1	9.9
50	1	5.3	7.5	4	9

5.3. Latency (ms)

In information exchange, latency refers to lag. It displays the duration required for data to travel through the system. Networks with increased delays reduce performance. Less delay ensures excellent transmission dependability. The latency results using the suggested and conventional systems are shown in Figure 7. The outcome of the latency is depicted in Table 3. It demonstrates that the recommended technique has lower latency than the existing strategies, allowing data to be transferred more effectively in the IoT and ES.

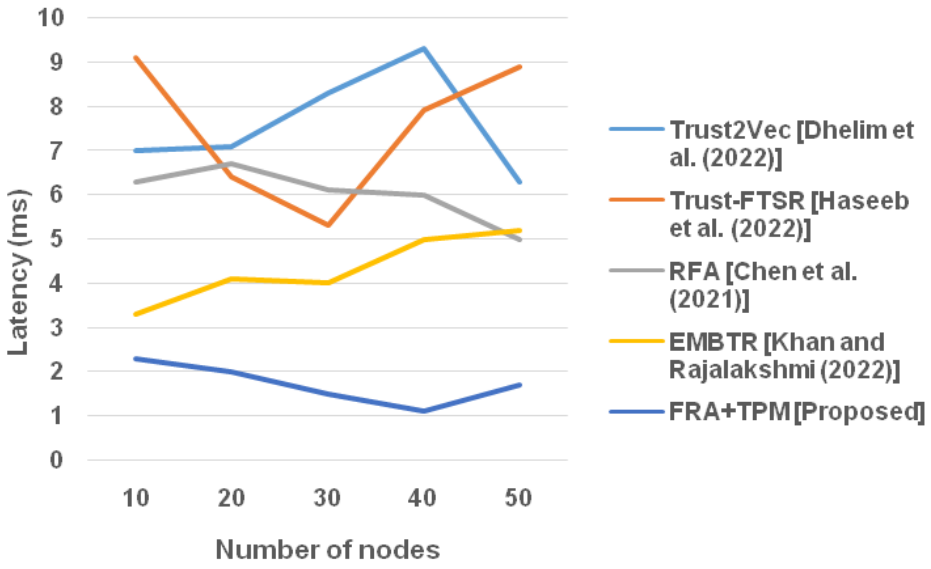

Figure 7. Latency for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 3. Outcome of latency.

Number of Nodes	Latency (ms)				
	Trust2Vec [24]	Trust-FTSR [25]	RFA [26]	EMBTR [27]	FRA + TPM [Proposed]
10	7	9.1	6.3	3.3	2.3
20	7.1	6.4	6.7	4.1	2
30	8.3	5.3	6.1	4	1.5
40	9.3	7.9	6	5	1.1
50	6.3	8.9	5	5.2	1.7

5.4. Energy Consumption (%)

The strategy's entire energy usage is referred to as energy consumption. It is the difference between the energy structure in its initial condition and the energy structure in its most recent state. The energy consumption results using the suggested and conventional systems are shown in Figure 8. The outcome of the energy consumption is depicted in Table 4. It is noticeable from Figure 7 that the recommended method requires less energy while computing. For interaction to be efficient, energy consumption must be maintained to a minimum.

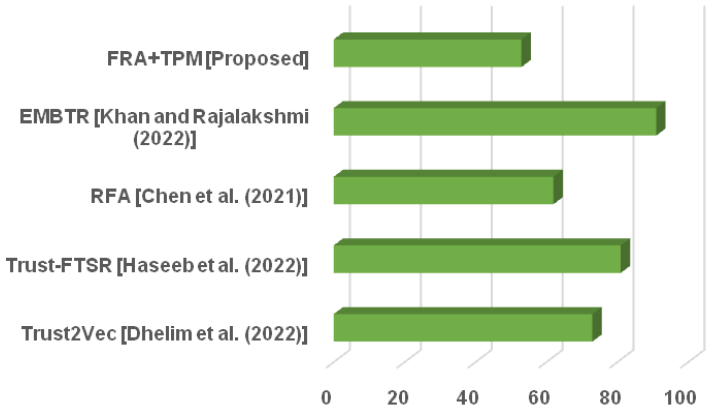

Figure 8. Energy consumption for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 4. Outcome of energy consumption.

Methods	Energy Consumption (%)
Trust2Vec [24]	73
Trust-FTSR [25]	81
RFA [26]	62
EMBTR [27]	91
FRA + TPM [Proposed]	53

5.5. Malignant Node Detection (%)

A malignant node attempts to prevent other nodes in the system from providing services to devices connected. The malignant node may reduce the system performance rate, which would reduce the network's service time. It is necessary to identify these nodes in wireless communication. The malignant node detection results using the suggested and conventional systems are shown in Figure 9. The outcome of the malignant node detection is depicted in Table 5. It shows that the provided approach effectively detects malicious nodes.

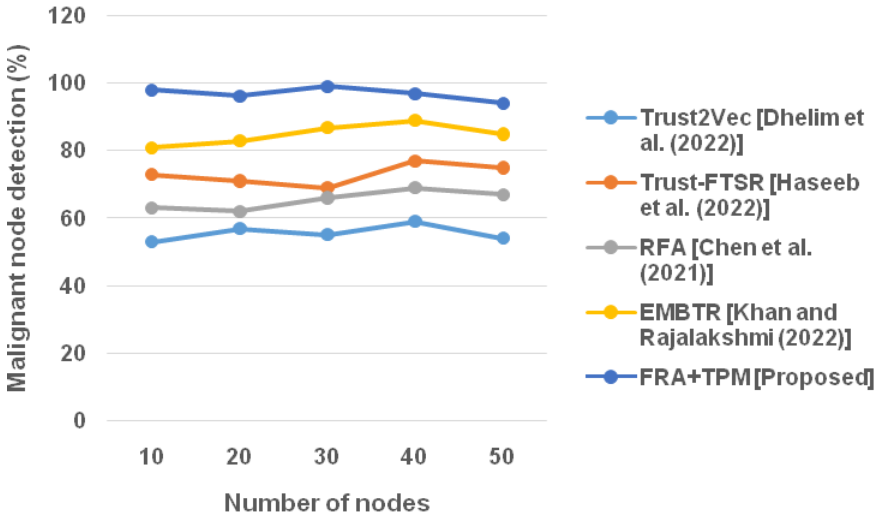

Figure 9. Malignant node detection for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 5. Outcome of malignant node detection.

Number of Nodes	Malignant Node Detection (%)				
	Trust2Vec [24]	Trust-FTSR [25]	RFA [26]	EMBTR [27]	FRA + TPM [Proposed]
10	53	73	63	81	98
20	57	71	62	83	96
30	55	69	66	87	99
40	59	77	69	89	97
50	54	75	67	85	94

5.6. Computation Time (s)

Computation time is the length of time required to accomplish a calculation (also known as execution time). This indicates how rapidly the modal can detect the result. The computation time results using the suggested and conventional systems are shown

in Figure 10. The outcome of the computation time is depicted in Table 6. As a result, the recommended method takes less time to forecast trust and find malignant nodes.

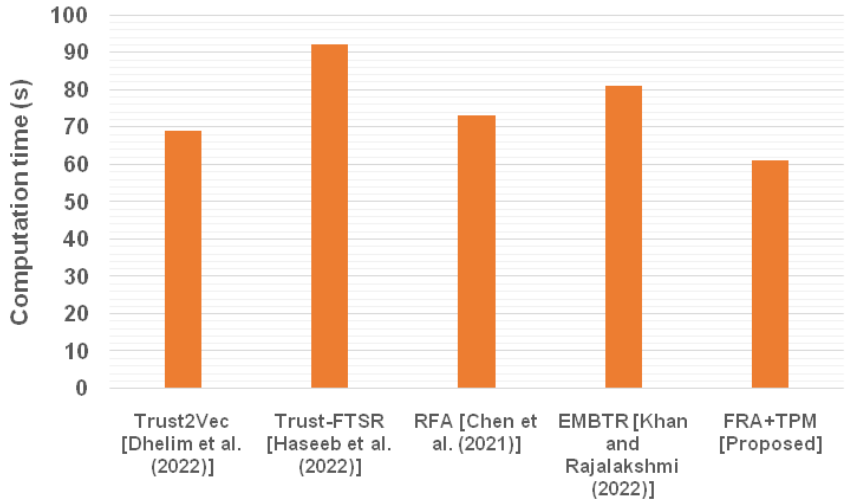

Figure 10. Computation time for suggested and conventional strategies i.e., Trust2Vec [24], Trust-FTSR [25], RFA [26] and EMBTR [27].

Table 6. Outcome of computation time.

Methods	Computation Time (s)
Trust2Vec [24]	69
Trust-FTSR [25]	92
RFA [26]	73
EMBTR [27]	81
FRA + TPM [Proposed]	61

6. Discussion

Dhelim et al. [24] presented the Trust2Vec TM technology, which could maintain connections in large-scale IoT networks and counter large-scale trust assaults carried out by many hostile units. They provided a network embeddings population identification technique that recognizes and inhibits communities of harmful nodes to identify assaults such as self-promotion and maligning. The accuracy of malicious detection is extremely low. To enhance trustworthy and cooperative connectivity in smart cities, the authors of [25] suggested the fault-tolerant supervised routing (Trust-FTSR) paradigm for TM in the IoT environment. For dependable and efficient system architecture, every node assesses the actions of its partners and develops mutual contact. A fault-tolerant transmitting mechanism was further provided by utilizing guided network learning without adding any extra overheads. The cost of computing is expensive. For computing immediate trust, using a gliding window and the time-decaying factor may significantly speed up the convergence rate. To efficiently screen out incorrect suggestions and reduce the impact of dangerous items, the authors of [26] designed a recommendation-filtering algorithm (RFA). As a means of adapting to the periodically dangerous situation, an adaptive weight was created to blend straight trust and suggestion trust into synthesis trust more effectively. The algorithm's complexity contributes to the computation's poor performance. The highest level of privacy is a constant need to protect the sensitive data sent between ESs and IoT. To complete the routing process safely based on nodes' trustworthy values, an enhanced multi-attribute-based attack resistance (EMBTR) method was suggested by the authors of [27]. By excluding the misbehaving nodes from the transmission channel based on the computed trusted measures for the network nodes, the method described in the study may offer a trustworthy route. The trust value forecasting process takes a long time to complete.

Owing to these shortcomings in the existing strategies, we offered an FRA+TPM for an effective TM in the IoT and ES.

7. Discussion of Analysis of the Presented Fuzzy Method with the Exact Method

One of the key advantages of the fuzzy method is its ability to handle uncertainty and imprecision. In IoT environments, where data and conditions can be uncertain or vague, the fuzzy approach allows for more flexible reasoning and decision making. In contrast, the exact method might struggle to handle uncertain or imprecise data effectively.

The fuzzy method provides a more granular and flexible approach to trust management. It allows for the definition of linguistic variables, membership functions, and fuzzy rules, enabling the system to reason and make decisions based on linguistic terms rather than rigid numerical values. This flexibility can be beneficial in capturing and expressing complex trust relationships. The exact method, on the other hand, may have more limited expressiveness and granularity due to its reliance on crisp values and strict rules.

The exact method may excel in situations where precision and deterministic decision making are crucial. However, in dynamic and uncertain IoT environments, the fuzzy method's ability to handle imprecision and uncertainty can contribute to increased robustness. The fuzzy method prioritizes adaptability and resilience to variations and changes in the environment, which may be advantageous in real-world IoT deployments.

It is worth considering the computational complexity of both methods. The exact method, depending on its mathematical or rule-based nature, may be computationally efficient but may lack the flexibility to adapt to changing conditions. On the other hand, the fuzzy method, particularly when involving complex fuzzy rule bases or large-scale systems, may require more computational resources for inference and decision making.

The fuzzy method offers unique advantages in handling uncertainty, providing flexibility, and achieving robustness in trust management within IoT-embedded systems. While the exact method may excel in certain scenarios that require precise decision making, the fuzzy method's ability to handle imprecision and uncertainty makes it particularly suitable for the complexities of IoT environments. The choice between the two methods depends on the specific requirements, context, and trade-offs desired in the trust management system.

8. Conclusions

IoT has the opportunity to become one of the best-known platforms in the age of online computation with the advancement in enabling capabilities and the invention of implementation methods. Any unit in the IoT platform that is capable of recognizing, detecting, connecting, and computing can interact with every other unit to achieve a specific goal using a variety of interaction styles. Embedded systems operate in more dynamic, unpredictable, and non-stationary ecosystems like those found in robot navigation and autonomous vehicle systems. As a result, there is an increase in requirement for their reliability, particularly when dealing with people. IoT device availability, performance efficiency, security, and sustainability issues have become greater due to the quick rise in IoT device utilization. To offer consumers improved services, these problems must be resolved. To manage trust effectively and increase IoT and ES's dependability, we proposed a hybrid fuzzy rule algorithm (FRA) and trust planning mechanism (TPM) (FRA+TPM). The proposed system was evaluated, and the findings are: trusted prediction (99%), energy consumption (53%), malignant node detection (98%), computation time (61 s), latency (1.7 ms), and throughput (9 Mbps). This proves that the proposed method is effective in TM in IoT and ES.

Based on the comprehensive work carried out in this study, the following management insights can be drawn:

- The integration of the FRA and trust planning mechanism offers enhanced trust management capabilities in IoT-embedded systems. By incorporating fuzzy logic, the approach effectively handles uncertainty, imprecision, and complex relationships among entities, enabling more accurate and adaptive trust assessment. This can

provide decision makers with valuable insights into the trustworthiness of IoT devices and facilitate informed decision making processes.

- The proposed approach's ability to adapt to changing conditions and handle uncertainties contributes to the robustness of trust management in dynamic IoT environments. The fuzzy-logic-based reasoning allows for flexible adjustments to evolving trust relationships and varying levels of trustworthiness, ensuring the system's resilience against environmental changes and potential threats.

Despite the rigorousness of the study, the following limitations need to be addressed:

- It is important to consider the computational complexity associated with the integration of FRA and trust planning mechanism. The process of fuzzy inference, rule evaluation, and defuzzification can introduce computational overhead, especially for large-scale IoT deployments or real-time applications. Further research and optimization techniques may be needed to mitigate potential performance issues.
- The proposed approach's applicability may vary depending on the specific characteristics and requirements of the IoT-embedded systems integration. The effectiveness of the FRA and trust planning mechanism may depend on the availability and quality of input data, the complexity of trust relationships, and the nature of the IoT environment. Thorough evaluation and validation of the approach in various scenarios would help assess its suitability for different contexts.
- The integration of the hybrid fuzzy rule algorithm and trust planning mechanism offers valuable insights into trust management in IoT-embedded systems. The approach enhances trust assessment, provides robustness in dynamic environments, and enables decision makers to make informed choices. However, the computational complexity and context-specific limitations should be considered when applying the proposed approach, highlighting the need for further research and validation to fully realize its potential in practical IoT deployments.

In further studies, the communication and effectiveness of the suggested TM mechanism will be optimized by various innovative techniques.

Author Contributions: N.V.R.R.—conceptualization, methodology, formal analysis, investigation, writing—original draft, P.P.—conceptualization, methodology, formal analysis, investigation, writing—original draft, M.M.—conceptualization, methodology, resources, writing—review and editing, S.S.—methodology, validation, writing—original draft, V.V.—methodology, validation, writing—original draft, V.T.—methodology, validation, writing—review and editing, L.C.—conceptualization, methodology, validation, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This article was supported by the project SP2023/074 Application of Machine and Process Control Advanced Methods supported by the Ministry of Education, Youth and Sports, Czech Republic.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lv, Z.; Song, H. Mobile Internet of Things Under Data Physical Fusion Technology. *IEEE Internet Things J.* **2019**, *7*, 4616–4624. [CrossRef]
2. Lv, Z.; Cheng, C.; Song, H. Digital Twins Based on Quantum Networking. *IEEE Netw.* **2022**, *36*, 88–93. [CrossRef]
3. Kumar, R.; Sharma, R. Leveraging blockchain for ensuring trust in IoT: A survey. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 8599–8622. [CrossRef]
4. Lv, Z.; Qiao, L.; Li, J.; Song, H. Deep-Learning-Enabled Security Issues in the Internet of Things. *IEEE Internet Things J.* **2020**, *8*, 9531–9538. [CrossRef]
5. Qu, Z.; Zhang, Z.; Liu, B.; Tiwari, P.; Ning, X.; Muhammad, K. Quantum detectable Byzantine agreement for distributed data trust management in blockchain. *Inf. Sci.* **2023**, *637*, 118909. [CrossRef]
6. Rana, K.; Singh, A.V.; Vijaya, P. Recent Trust Management Models for Secure IoT Ecosystem. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 23–33.

7. Lv, Z.; Wu, J.; Li, Y.; Song, H. Cross-Layer Optimization for Industrial Internet of Things in Real Scene Digital Twins. *IEEE Internet Things J.* **2022**, *9*, 15618–15629. [CrossRef]
8. Dai, X.; Xiao, Z.; Jiang, H.; Alazab, M.; Lui, J.C.S.; Dustdar, S.; Liu, J. Task Co-Offloading for D2D-Assisted Mobile Edge Computing in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 480–490. [CrossRef]
9. Liu, Y.; Yu, W.; Rahayu, W.; Dillon, T. An Evaluative Study on IoT ecosystem for Smart Predictive Maintenance (IoT-SPM) in Manufacturing: Multi-view Requirements and Data Quality. *IEEE Internet Things J.* **2023**, 1. [CrossRef]
10. Babaei, A.; Khedmati, M.; Jokar, M.R.A.; Tirkolaee, E.B. Designing an integrated blockchain-enabled supply chain network under uncertainty. *Sci. Rep.* **2023**, *13*, 1–19. [CrossRef]
11. Mazumdar, H.; Kaushik, A.; Gohel, H.A. To mitigate primary user emulation attack trajectory using cognitive single carrier frequency division multiple access approaches: Towards next generation green IoT. *Eng. Rep.* **2023**, e12672. [CrossRef]
12. Mishra, N.; Pandya, S. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access* **2021**, *9*, 59353–59377. [CrossRef]
13. Praveena, J.; Arivazhagan; Reddy, P.V.P. Blockchain based sensor system design for embedded IoT. *J. Comput. Inf. Syst.* **2023**, 1–18. [CrossRef]
14. Din, I.U.; Bano, A.; Awan, K.A.; Almogren, A.; Altameem, A.; Guizani, M. LightTrust: Lightweight Trust Management for Edge Devices in Industrial Internet of Things. *IEEE Internet Things J.* **2021**, *10*, 2776–2783. [CrossRef]
15. Bitencourt, H.V.; de Souza, L.A.F.; Santos, M.C.D.; Silva, R.; de Lima e Silva, P.C.; Guimarães, F.G. Combining embeddings and fuzzy time series for high-dimensional time series forecasting in internet of energy applications. *Energy* **2023**, *271*, 127072. [CrossRef]
16. Shi, P.; Wang, H.; Yang, S.; Chen, C.; Yang, W. Blockchain-based trusted data sharing among trusted stakeholders in IoT. *Softw. Pract. Exp.* **2019**, *51*, 2051–2064. [CrossRef]
17. Fahim, M.; El Mhouti, A.; Boudaa, T.; Jakimi, A. Modeling and implementation of a low-cost IoT-smart weather monitoring station and air quality assessment based on fuzzy inference model and MQTT protocol. *Model. Earth Syst. Environ.* **2023**, 1–18. [CrossRef]
18. Diwan, T.D.; Choubey, S.; Hota, H.S.; Goyal, S.B.; Jamal, S.S.; Shukla, P.K.; Tiwari, B. Feature Entropy Estimation (FEE) for Malicious IoT Traffic and Detection Using Machine Learning. *Mob. Inf. Syst.* **2021**, *2021*, 1–13. [CrossRef]
19. Abawajy, J.; Darem, A.; Alhashmi, A.A. Feature Subset Selection for Malware Detection in Smart IoT Platforms. *Sensors* **2021**, *21*, 1374. [CrossRef]
20. Zheng, W.; Deng, P.; Gui, K.; Wu, X. An Abstract Syntax Tree based static fuzzing mutation for vulnerability evolution analysis. *Inf. Softw. Technol.* **2023**, *158*, 107194. [CrossRef]
21. Gong, J.; Rezaeipanah, A. A fuzzy delay-bandwidth guaranteed routing algorithm for video conferencing services over SDN networks. *Multimed. Tools Appl.* **2023**, 1–30. [CrossRef] [PubMed]
22. Rey, V.; Sánchez, P.M.S.; Celdrán, A.H.; Bovet, G. Federated learning for malware detection in IoT devices. *Comput. Netw.* **2022**, *204*, 108693. [CrossRef]
23. Alkahtani, H.; Aldhyani, T.H.H. Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. *Sensors* **2022**, *22*, 2268. [CrossRef] [PubMed]
24. Dhelim, S.; Aung, N.; Kechadi, M.T.; Ning, H.; Chen, L.; Lakas, A. Trust2Vec: Large-Scale IoT Trust Management System Based on Signed Network Embeddings. *IEEE Internet Things J.* **2022**, *10*, 553–562. [CrossRef]
25. Haseeb, K.; Saba, T.; Rehman, A.; Ahmed, Z.; Song, H.H.; Wang, H.H. Trust Management With Fault-Tolerant Supervised Routing for Smart Cities Using Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 22608–22617. [CrossRef]
26. Chen, G.; Zeng, F.; Zhang, J.; Lu, T.; Shen, J.; Shu, W. An adaptive trust model based on recommendation filtering algorithm for the Internet of Things systems. *Comput. Netw.* **2021**, *190*, 107952. [CrossRef]
27. Khan, A.F.; Rajalakshmi, C.N. A multi-attribute based trusted routing for embedded devices in MANET-IoT. *Microprocess. Microsyst.* **2022**, *89*, 104446. [CrossRef]