IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 May 2023, accepted 7 June 2023, date of publication 9 June 2023, date of current version 16 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284677

RESEARCH ARTICLE

Securing IoT With Deep Federated Learning: A Trust-Based Malicious Node Identification Approach

KAMRAN AHMAD AWAN[1], IKRAM UD DIN[1], (Senior Member, IEEE), MAHDI ZAREEI[2], (Senior Member, IEEE), AHMAD ALMOGREN[3], (Senior Member, IEEE), BYUNG SEO-KIM[4], (Senior Member, IEEE), AND JESÚS ARTURO PÉREZ-DÍAZ[2], (Member, IEEE)

[1]Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan
[2]Tecnologico de Monterrey, School of Engineering and Sciences, Zapopan 45201, Mexico
[3]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia
[4]Department of Software and Communications Engineering, Hongik University, Sejong 30016, South Korea

Corresponding authors: Mahdi Zareei (m.zareei@ieee.org) and Jesús Arturo Pérez-Díaz (jesus.arturo.perez@tec.mx)

This work was supported in part by the Tecnologico de Monterrey and in part by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project under Grant RSP2023R184.

ABSTRACT The Internet of Things (IoT) has revolutionized the world with its diverse applications and smart connected devices. These IoT devices communicate with each other without human intervention and make life easier in many ways. However, the independence of these devices raises several significant concerns, such as security and privacy preservation due to malicious and compromised nodes within the network. Trust management has been introduced as a less computationally intensive alternative to traditional approaches such as cryptography. The proposed FedTrust approach addresses these challenges by designing a method for identifying malicious and compromised nodes using federated learning. FedTrust trains edge nodes with a provided dataset and forms a global model to predict the abnormal behavior of IoT nodes. The proposed approach utilizes a novel trust dataset consisting of 19 trust parameters from three major components: knowledge, experience, and reputation. To reduce the computational burden, FedTrust employs the concept of communities with dedicated servers to divide the dataset into smaller parts for more efficient training. The proposed approach is extensively evaluated in comparison to existing approaches in terms of accuracy, precision, and other metrics to validate its performance in IoT networks. Simulation results demonstrate the effectiveness of FedTrust by achieving a higher rate of detection and prediction of malicious and compromised nodes.

INDEX TERMS Internet of Things, federated learning, trust management, deep learning, malicious nodes, security, privacy preservation, trustworthiness.

I. INTRODUCTION

The Internet of Things (IoT) [1] is an emerging concept where everyday objects are connected to the internet. The interconnection allows these devices to share data independently over the Internet. This has caused a profusion of new services and applications, which is altering how we work and live. However, the security and privacy of the data produced by these devices raise vulnerabilities due to independent connectivity [2]. To maintain the secure and dependable operation of IoT networks, it is imperative to solve crucial issues including the security and privacy of sensitive data and the integrity of the communication performed between devices. In order to protect IoT networks, trust management [3] is essential

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu.

since it guarantees the dependability and legitimacy of the participating devices and their data. A secure communication route between devices and the prevention of malevolent devices from jeopardizing network security are both made possible by trust management. Detection of malevolent nodes [4], that can compromise network security by engaging in malicious activities including eavesdropping [5], tampering with data [6], and launching denial-of-service attacks [7], is one of the primary issues in trust management for IoT networks [8]. Numerous trust management strategies, including reputation-based systems [9], cryptographic methods [10], and machine learning-based methods, have been proposed as solutions to these research challenges.

The majority of the existing methods of managing trust in IoT contexts frequently include aspects of Blockchain and federated learning [11]. Through federated learning, many nodes can work together to build a machine-learning model [12] without sharing any of their private data. To achieve this, the model is trained using locally aggregated gradients, which are then added together to update the shared model. This approach not only assures that the model is trained on a more varied set of data, but it also improves the privacy and efficiency of the model. Furthermore, Blockchain technologies offer a safe and impenetrable ledger of all network transactions [13]. Blockchain can be used in an IoT setting to keep track of all device interactions [14], including the execution of smart contracts and the transfer of assets [15]. It is possible to create a transparent and safe trust management mechanism that can provide robust security against attacks [16] by integrating Blockchain into trust management systems. However, utilizing Blockchain in IoT environment may increase the security but also increases the computational complexity due to which becomes difficult to implement for less-capable nodes.

Most current approaches utilize machine learning, federated learning, and blockchain technologies to maintain trustworthiness, as discussed in Section II. Because of the widespread significance of IoT devices, it is vital that they be safeguarded against compromised and malicious nodes. Standard security measures, such as encryption, may not be suitable for resource-constrained IoT systems due to their computationally expensive nature. Because of these shortcomings, current approaches to IoT trust management are vulnerable to attacks such as whitewashing and bad-mouthing. A more robust, private, and scalable approach to protecting IoT networks from malicious and compromised nodes may be achieved by combining deep learning with federated learning. While the use of these approaches may provide security, their implementation on a large scale [17] becomes impossible, making them unsuitable for real-world IoT solutions. In the proposed approach, we address these challenges by reducing computational complexity through the use of communities and domains. The proposed approach modifies federated learning by dividing the dataset into chunks controlled by dedicated servers as illustrated by Figure 1, which then allocate smaller portions of the dataset to IoT nodes for training purposes. This reduces computational complexity and increases device efficiency. The major contribution of the proposed approach can be summarized as:

1) The proposed technique utilizes Deep Federated Learning to identify malicious and compromised nodes based on trust management in IoT. The FedTrust combines the strengths of both Federated Learning and Deep Learning to efficiently and accurately identify malicious and compromised nodes.
2) The proposed technique addresses the limitations of traditional trust management techniques in IoT, which can be vulnerable to attacks such as whitewashing and bad-mouthing by using a novel dataset consisting of trust parameters to predict the abnormal behavior of nodes.
3) The proposed technique also reduces the extensive capabilities required for training purposes and optimizes the process to make it efficient for the Green IoT environment and makes real-time security.

The structure of the rest of the article is as follows: Section II discusses existing approaches, elaborates on their contributions and limitations. Section III provides an extensive discussion of the proposed methodology, dataset, splitting, and model training. Section IV illustrates a comparative analysis of the proposed approach with existing approaches in terms of metrics such as Precision, Accuracy, and F1 Score. Finally, Section V concludes the article.

II. BACKGROUND STUDY

The identification of such nodes that can affect the trustworthiness of the IoT environment is a notable challenge. Several existing approaches use traditional methods to maintain trustworthiness but are unable to adequately fulfilled the need by efficiently predicting the behavior while maintaining low energy consumption. This section illustrates the existing approaches and identifies the limitation of those approaches.

In [18], a decentralized-based trust management approach is proposed that utilizes blockchain and federated learning to maintain trustworthiness in an IoT environment. The proposed approach protocol consists of trust scores, trust deviation, and trust consistency values to perform computations in the blockchain. The proposed architecture consists of an aggregator, selected group leaders, and a coalition group. The major contribution of the proposed mechanism is the utilization of federated learning to train the decentralized nodes, however, required extensive computational capabilities to perform computations in the blockchain that may reduce throughput [19]. In [20], another trust-driven approach is proposed that uses a reinforcement selection strategy with a double deep Q-Learning algorithm [21] along with federated learning for the identification of malicious and compromised nodes. The proposed model consists of an edge server that process and creates the global model, the weights uploading

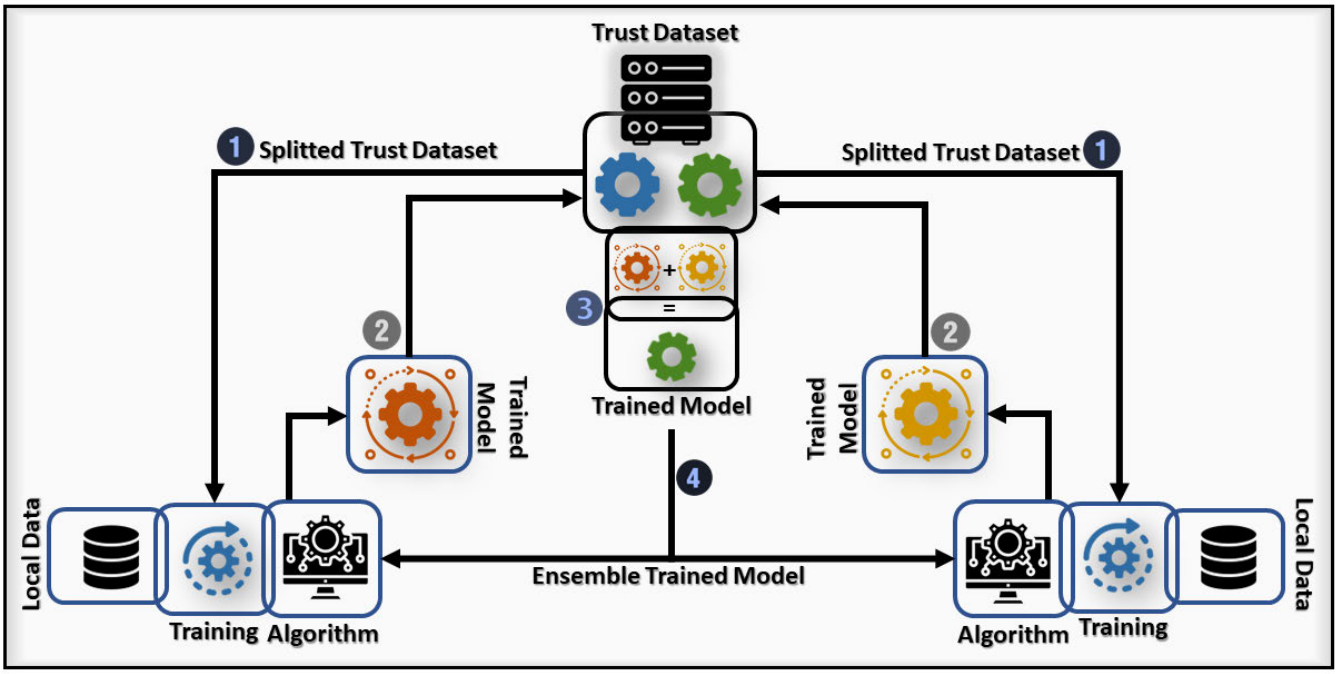


FIGURE 1. The training process of the deep federated learning model.

and model updating process, and the local layers in which IoT devices are used to train the model. The significant contribution of the proposed mechanism is the utilization of Q-Learning along with selection strategy, and federated learning. However, the proposed is computationally intensive and can raise the overhead ratio to complete the computation in real-time [22].

In [23], a trust-augmented-based deep reinforcement learning approach is proposed for client selection using federated learning. Along with the client selection component, the proposed approach also utilizes transfer learning to handle the data and to address the vulnerabilities caused by low efficient training. The working architecture of the approach consists of three major layers which are the edge cloud layer, IoT layer, and public environment. The edge cloud layer performs the knowledge transfer to the aggregation server for global and local model formulation. The IoT layer consists of IoT devices that locally trained the model. Furthermore, the public environment consists of individuals for COVID-19 [24] detection using camera sensors. In [25], a hierarchical-based blockchain framework is proposed for IoT intrusion detection using federated learning. The data flow in the proposed approach consists of three major components i.e., global model aggregator, local model aggregator, and IoT edge nodes to locally train the model. The communication among these nodes is performed for two major purposes which are uploading model weights and downloading aggregated model weights.

Another trust-based approach is proposed to aggregate the trusted features using federated learning for attack detection [26]. The approach stores the training process of the model on the blockchain whereas the intrusion alarm is set to the cloud for global prediction. The proposed approach architecture consists of four layers i.e., the Blockchain layer, chain-code layer, federated layer, and application layer. The significant contribution of the proposed approach is the filter the false alerts using semi-supervised learning. However, the approach may face issues to handle intrusion accurately due to bandwidth limitations. In [27], an approach is proposed to maintain end-to-end device trustworthiness by securing the IoT infrastructures with federated learning and Blockchain. The proposed model consists of 4 major components i.e., cloud, blockchain, fog server, and edge nodes. The trusted devices are appended to the Blockchain whereas fog performs the federated aggregation process received by the edge IoT nodes.

III. PROPOSED FedTrust METHODOLOGY

The proposed FedTrust approach utilizes the concept of federated learning and communities to efficiently maintain trustworthiness. The proposed approach trained the model using edge nodes with modified federated learning implementation architecture. This section will elaborate on the proposed approach architecture, along with dataset features, splitting, and the training process of deep federated learning. Federated learning is used in our approach, which frees edge nodes from sharing data as they train their models. Unlike with some other available options, this one guarantees that no private data will be disclosed. The proposed solution is well-suited to dealing with large-scale IoT networks because

TABLE 1. Analysis of current state-of-the-art methods.

Ref.	Approach	Technique	Application
[18]	Decentralized Trust Management	Blockchain-empowered Federated Learning	IoT Security
[20]	Trust-driven Reinforcement Selection	Federated Learning	IoT Devices
[22]	Application or Infrastructure Co-design	Real-time Edge Video Analytics	IoT Analytics
[23]	Trust-augmented Deep Reinforcement Learning	Federated Learning Client Selection	IoT Networks
[25]	Hierarchical Blockchain-based FL	Collaborative IoT Intrusion Detection	IoT Security
[26]	Trusted Feature Aggregator FL	Malicious Attack Detection	IoT Security
[27]	Blockchain-supported FL	Securing Critical IoT Infrastructures	IoT Security

to deep federated learning's ability to distribute the computational burden across edge nodes. This saves a lot of time and money compared to traditional, centralized educational models. Defense against external forces: Our novel trust dataset with trust parameters makes the proposed approach more resilient to common attacks like whitewashing and bad-mouthing. We will demonstrate how FedTrust performs in comparison to other solutions under various forms of assault. Thanks to its ability to continuously learn and adapt to the changing behavior of IoT nodes, our technique is particularly well-suited for dynamic and heterogeneous IoT networks.

A. OVERVIEW OF THE PROPOSED TECHNIQUE

The proposed approach is based on the modified implementation of federated learning by employing the domain server that handles the communities and training the model by dividing the dataset. The proposed approach architecture consists of four major components, i.e., global dataset, global model, domain server, and communities that contain edge IoT nodes as illustrated by Figure 2.

The global dataset in the proposed model consists of several distinct features of trust that are gathered based on three trust components which are knowledge, reputation, and experience whereas the detail of the dataset is illustrated in Section III-B. The global dataset is split into several parts based on the available domain servers and delivered to each domain server. Further, the domain server selected the nodes from the communities based on the capabilities, and competency to train the received global dataset. After that, the domain server further split the dataset into the number of selected nodes and each part to the specific nodes for training purposes. After the training process, these nodes transmit the trained model to the domain server where the domain server merges all the received datasets and labeled them as the communities train model, and transmits it back to the central server to merge all the communities train model and formulate the global model.

The formulated global dataset is then transmitted back to the domain server and then the trained model will be shared with the edge nodes. Another major aspect of the proposed mechanism is that the dataset splitting and training is a time-driven process and it performs after a pre-defined time interval so that the model gets trained with real-time changes to maintain robustness. Furthermore, the splitting of dataset by the domain server will help to minimize the computational burden on the edge nodes which will also have a significant impact to minimize energy consumption. As the

Algorithm 1 The Working Flow of the Trust Assessment in IoT for Node n

Input: Node n, weights ω_1, ω_2, ω_3, α, τ, ζ, υ_1, υ_2, and w_i
d **Output:** Trust value T for node n
Knowledge Component:
Normalize the parameters using Equation 23.
Assign weights W_i to the parameters.
Compute the knowledge metric K using Equation 10 - 16.
Reputation Component:
Collect feedback and ratings (P_f, N_f), social proof (E_p, N_n), transparency rating (TI, TI_{max}), data breaches and security vulnerabilities (DB_n, SV_n, DB_{max}, SV_{max}), user engagement (I_i, n), and responsiveness of node (RT, R_h) for node n.
Compute the reputation metric parameters using Equations 17 - 22.
Normalize the parameters using Equation 23.
Assign weights W_i to the parameters.
Compute the reputation metric R using Equation 24.
Experience Component:
Collect interaction frequency (n, T_i, w_i), transaction success rate (S_{ti}, v_i, T_{ti}), time taken to complete transactions (t_i, u_i), communication quality (SNR, BER, α), data sharing behavior (p_{ij}), cooperation level (C_n, C_{max}), and end-to-end packet delivery (PDR_i, z) for node n.
Compute the experience metric parameters using Equations 25 - 31.
Normalize the parameters using Equation 23.
Assign weights W_i to the parameters.
Compute the experience metric E using Equation 32.
Aggregating the Metrics:
Compute the trust value T for node n using Equation 32 and the weights ω_1, ω_2, and ω_3 to combine the knowledge metric K, the reputation metric R, and the experience metric E.

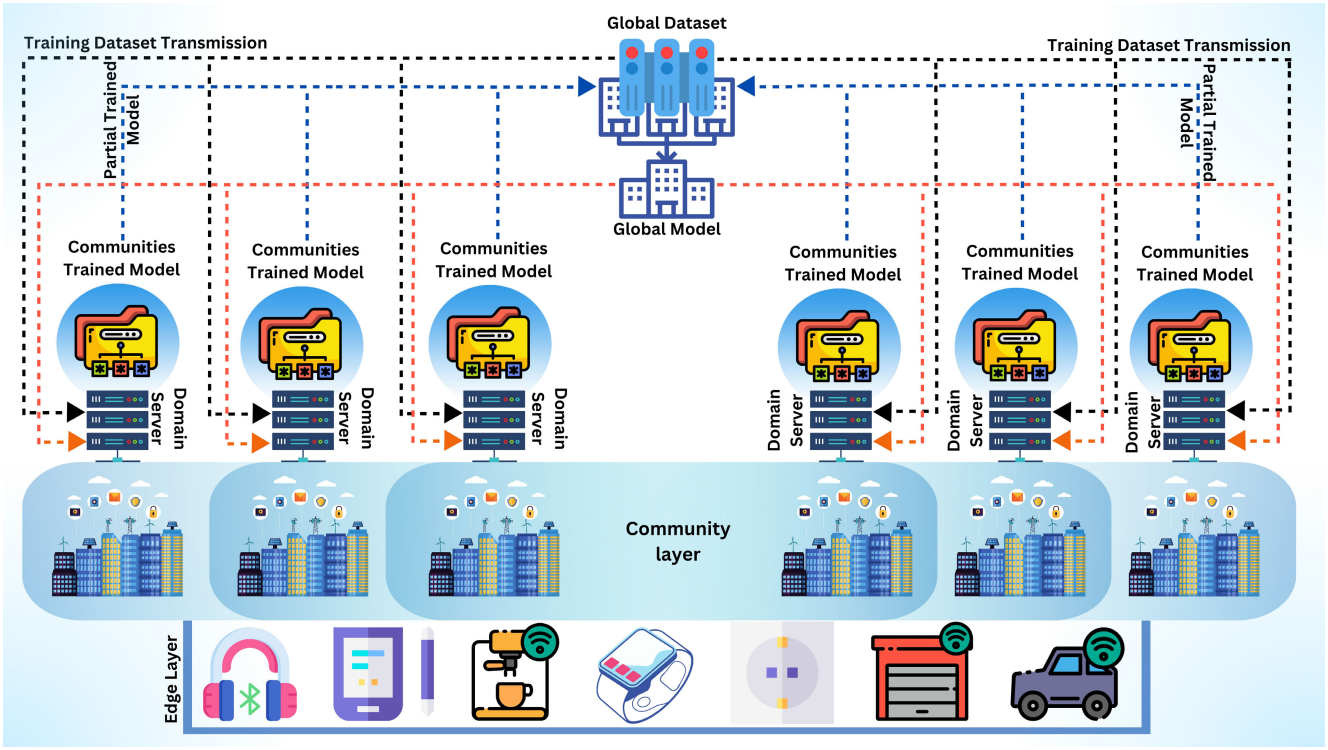


FIGURE 2. The proposed FedTrust working architecture.

splitting is performed by the central and domain server also has a notable impact to optimize the training process.

B. TRUST MANAGEMENT MATRICES

The dataset of the proposed approach is comprised of distinct features of trust components whereas these features are selected to provide nodes with the intelligence to predict the behavior as well as the patterns of different IoT potential attacks to maintain trustworthiness. The dataset is consist of 90788 values of each parameter with multiple observations. This section extensively explains the features utilizes in the training process and the mathematical computational model that has been utilized in the creation of the dataset whereas the complete working flow of the proposed mechanism is illustrated by Algorithm 1.

1) KNOWLEDGE METRIC

The knowledge component of trust is an important element of node trustworthiness because it determines nodes' confidence in neighboring nodes. The knowledge component of trust in the context of IoT networks refers to nodes' understanding of the network's and connected devices' security and privacy. To establish and maintain a high level of trustworthiness in an IoT network, it is critical that users have access to reliable and up-to-date knowledge about the network's privacy and security measures. The knowledge of the nodes is based on prior observations that will help in the identification of whitewashing attacks. In this trust component, a novel combination of seven trust parameters has been implemented which are credibility, accuracy, reliability, compliance, capabilities, availability, and responsiveness. In creation of the data, the first step is to initialize variables for the trust parameters with default values set to 0.0. Second step is to calculate the credibility of the node provider by evaluating its expertise. Third, the evaluate of the security and privacy of the node is calculated using accuracy and reliability. The accuracy and reliability is calculated using performance metrics and data validation methods. Fourth, the assess level of compliance is calculated by evaluating the node against established standards such as ISO and NIST. Each parameter of knowledge components is evaluated and calculated as:

- To describe a node's credibility (C), we may add together its previous performance (P), its expertise (E), and a time decay factor ($delta$) to account for the credibility drop over time:

$$C = \alpha P + \beta E + \gamma \delta^t \tag{1}$$

  where $alpha$, $beta$, and $gamma$ are weights adding up to 1 that represent the relative relevance of track record, competence, and the decaying influence of time, respectively.
- Taking into account a weight factor ($omega$) to regulate the balance between precision and recall, accuracy (A) may be computed as the harmonic mean of

precision (P_r) and recall (R_c):

$$A = \frac{(1+\omega^2)\cdot P_r \cdot R_c}{(\omega^2 \cdot P_r) + R_c} \tag{2}$$

- Combining a node's performance stability (T) with a confidence factor (CF) that takes into account the node's historical performance variation yields reliability (R):

$$R = \frac{\sum_{t=1}^{T} P_t}{T} \cdot CF \tag{3}$$

where $CF = \frac{1}{1+e^{-\lambda(\frac{\sum_{t=1}^{T} P_t}{T}-\mu)}}$ is a sigmoid function that considers the average performance and a threshold (μ) with a steepness factor (λ).

- One way to express compliance (Com) is as a weighted sum of the node's adherence to best practices (BP) and the number of standards attained ($S_achieved$)

$$Com = \frac{\eta S_{achieved} + (1-\eta)BP}{S_{total}} \tag{4}$$

where eta is a weight between 0 and 1 that indicates how significant meeting standards and following best practices really are.

- The capabilities of a node (Cap), when weighted appropriately, are the sum of its computing power (CP), storage capacity (SC), and energy efficiency (EE)

$$Cap = \theta_1 \frac{CP}{CP_{max}} + \theta_2 \frac{SC}{SC_{max}} + \theta_3 \frac{EE}{EE_{max}} \tag{5}$$

Maximum values of CP, SC, and EE are denoted by CP_max, SC_max, and EE_max, respectively; the weights $theta_1$, $theta_2$, and $theta_3$ add up to 1, showing the significance of each factor.

- Assuming a decay factor ($delta_av$) that compensates for the decline in availability over time, Availability (Av) may be determined as a function of the Uptime (U) of the node and the Total Time (T_total) of the Network:

$$Av = \frac{U}{T_{total}} \cdot e^{-\delta_{av} t} \tag{6}$$

- Response time (RT) and requests handled (R_h) may be combined to describe responsiveness (Res), with a weighted value assigned to each component.

$$Res = \frac{\phi_1}{1+e^{k_1(RT-RT_{th})}} + \phi_2 \frac{R_h}{R_{max}} \tag{7}$$

where RT_th is the response time threshold, k_1 is the steepness factor, and R_max is the maximum number of requests handled by the network; the weights phi_1 and phi_2 add up to 1, reflecting the proportional significance of each component.

Let's say that C stands for credibility, A for accuracy, R for reliability, Com for compliance, Cap for capabilities, Av for availability, and Res for responsiveness, and so on. Following is a working definition of the component of knowledge:

$$K = f(C, A, R, Com, Cap, Av, Res) \tag{8}$$

The following are the stages of data creation:

- Trust parameter variables should be initialized at 0.0.

$$C = A = R = Com = Cap = Av = Res = 0.0 \tag{9}$$

- Determine the node provider's trustworthiness (in C) by its level of competence. A node's reputation is equal to the product of its previous performance (P) and its expertise (E), with P being more heavily weighted than E.

$$C = \alpha P + \beta E \tag{10}$$

where $alpha$ and $beta$ are weights adding up to 1, showing the significance of proven results and specialized knowledge in establishing trustworthiness.

- Assess the node's privacy and security using a measure of precision (A) and trustworthiness (R). The accuracy can be determined by dividing the number of right decisions by the total number of choices ($D_correct/D_total$), and the reliability can be determined by measuring the stability of a node's output over time (T):

$$A = \frac{D_{correct}}{D_{total}} \tag{11}$$

$$R = \frac{\sum_{t=1}^{T} P_t}{T} \tag{12}$$

- Determine the node's compliance (Com) by measuring it against industry benchmarks like ISO and NIST guidelines. A higher score implies more compliance, and this may be expressed as a compliance score between 0 and 1:

$$Com = \frac{S_{achieved}}{S_{total}} \tag{13}$$

in where $S_achieved$ is the total number of met standards and S_total is the total number of standards.

Assess the node's privacy and security using a measure of precision (A) and trustworthiness (R). The accuracy can be determined by dividing the number of right decisions by the total number of choices ($D_correct/D_total$), and the reliability can be determined by measuring the stability of a node's output over time (T):

$$A = \frac{D_{correct}}{D_{total}} \tag{14}$$

$$R = \frac{\sum_{t=1}^{T} P_t}{T} \tag{15}$$

Determine the node's compliance (Com) by measuring it against industry benchmarks like ISO and NIST guidelines. A higher score implies more compliance, and this may be expressed as a compliance score between 0 and 1:

$$Com = \frac{S_{achieved}}{S_{total}} \tag{16}$$

in where $S_achieved$ is the total number of met standards and S_total is the total number of standards.

2) REPUTATION METRIC

The reputation component of trust in the IoT environment refers to the evaluation of node's trust degree based on its previous behaviour and performance in the network being recognized by neighboring nodes. This evaluation is critical for the secure and efficient operation in IoT environment because it provides equal opportunity in comparison with the knowledge components for the identification of malicious and compromised nodes and prevents them from interfering with network operations. The reputation parameter dataset consists of several distinct trust parameters i.e., feedback and ratings from previous node/devices, social proof by the provider node, level of transparency rating related to functioning, data breaches incidence level and security vulnerabilities, user engagement, responsiveness of node. The parameters of the reputation measure are calculated and implemented as described below:

- Positive feedback (P_f) is more weighted than negative feedback (N_f), hence the total of the two may be used to indicate ratings and feedback (F):

$$F = \frac{\omega_1 P_f - \omega_2 N_f}{P_f + N_f} \tag{17}$$

  When the positive and negative feedback are given equal weight by the sum of their respective weights, $omega_1$ and $omega_2$.
- To determine social proof (SP), divide the number of nodes endorsing the provider node (E_p) by the number of nodes in the immediate vicinity (N_n) and multiply by a weighting factor.

$$SP = \frac{\tau E_p}{N_n} \tag{18}$$

  where tau is a weight between 0 and 1 indicating how much reliance one should place on social evidence when assessing reputation.
- The Transparency Rating (TR) is equal to the maximum Transparency Index (TI_max) divided by the transparency index (TI), which measures the openness of the node's operation and data sharing.

$$TR = \frac{TI}{TI_{max}} \tag{19}$$

- Considering the relative relevance of data breaches (DB_n) and security vulnerabilities (SV_n), the total number of data breaches and security vulnerabilities (DB) may be expressed as a weighted sum of the two quantities.

$$DB = \frac{\upsilon_1 DB_n + \upsilon_2 SV_n}{DB_{max} + SV_{max}} \tag{20}$$

  where the highest values of the parameters in the network are denoted by DB_max and SV_max, and the weights $upsilon_1$ and $upsilon_2$ add up to 1, showing the relative significance of each factor.
- Average interactions (I) per node may be used as a proxy for user engagement (UE).

$$UE = \frac{\sum_{i=1}^{n} I_i}{n} \tag{21}$$

  for every given size n of network nodes.
- Node responsiveness (RN) may be expressed as the harmonic mean of response time (RT) and requests handled (R_h), with a weight factor ($zeta$) regulating the relative importance of these two metrics.

$$RN = \frac{(1+\zeta^2)\cdot \frac{1}{RT}\cdot R_h}{\zeta^2 \cdot \frac{1}{RT} + R_h} \tag{22}$$

The formulation of a dataset starts with a numerical representation of the parameters, having values between 0 and 1. Parameters are normalized by calculating their normalized values, or NR_i; for each parameter, R_i, as illustrated in Equation 23:

$$NR_i = (R_i - R_{min})/(R_{max} - R_{min}) \tag{23}$$

In Equation 23, R_{min} and R_{max} are the minimum and maximum possible values for R_i, respectively. In the next phase, weight is allocated to each parameters of the reputation components whereas the assign weights (W_i) ranges from 0 to 1 and the sum of all weights is 1. Further, a computation is performed to evaluate the reputation (R) as illustration by Equation 24.

$$R = \sum (\omega_i * NR_i) \tag{24}$$

3) EXPERIENCE METRIC

The experience component of trust parameter is the most significant as it contain those ratings that are provided by the user after the completion of the task. The proposed approach dataset implements several parameters to perform effectively and identify malicious and compromised nodes with enhanced prediction rate. The experience components trust parameters are: (i) interaction frequency (ii) transaction success rate (TSR) (iii) time taken to complete transactions (iv) Communication quality (v) Resource utilization (vi) Data sharing behavior (v) Cooperation level (vi) end-to-end packet delivery.

In the proposed approach, the interaction frequency computation is performed by counting the number of interactions between two nodes over a certain period of time. The equation used to perform the computation is by using the number of interactions n by the total time period T. The TSR parameter is computed by dividing the number of successful transactions by the total number of transactions. The time taken to complete transactions is implemented by measuring the average time taken to complete all transactions. The communication quality in the dataset creation is implemented by measuring the quality of communication between two nodes. The data sharing behavior parameter is implemented and performs the computation by analyzing the behavior of nodes in terms of data sharing. The cooperation level parameter is measured

by analyzing the cooperation level between two nodes during interactions. The End-to-End packet delivery is computed by measuring the delivery ratio of packets between two nodes. Listed below are descriptions of the specific ways in which the suggested method makes use of experience metric parameters:

- The number of interactions (n) between two nodes across many time intervals ($T_1, T_2, cdots, T_k$) is averaged and then weighted to get the interaction frequency (IF):

$$IF = \frac{\sum_{i=1}^{k} w_i n_i}{\sum_{i=1}^{k} w_i T_i} \tag{25}$$

  where w_i is the value used to quantify the significance of each interval.
- The Transaction Success Rate (TSR) is calculated by averaging the success rates of individual transactions ($S_t1, S_t2, cdots, S_tp$) based on their relative importance.

$$TSR = \frac{\sum_{i=1}^{p} v_i S_{ti}}{\sum_{i=1}^{p} v_i T_{ti}} \tag{26}$$

  where v_i is the value allocated to transaction type i and T_ti is the sum of all transactions of type i.
- As a weighted harmonic mean of the times it takes to complete each transaction, the Time Taken to Complete Transactions ($TTCT$) is determined.

$$TTCT = \frac{\sum_{i=1}^{m} u_i}{\sum_{i=1}^{m} \frac{u_i}{t_i}} \tag{27}$$

  where u_i is the value placed on transaction i and t_i is the time it took to finish i.
- Signal-to-noise ratio (SNR) and bit error rate (BER) are used to calculate communication quality (CQ), which is then weighted geometrically.

$$CQ = \sqrt[\alpha]{SNR^{\alpha}(1 - BER)^{1-\alpha}} \tag{28}$$

  where $alpha$ is a non-negative integer weighting factor.
- Normalized mutual information (NMI) between the shared data and the whole data is used to quantify data sharing behavior (DSB):

$$DSB = \frac{\sum_{i=1}^{r} \sum_{j=1}^{c} p_{ij} \log \frac{p_{ij}}{p_{i\cdot} p_{\cdot j}}}{\sqrt{H(D_s)H(D_t)}} \tag{29}$$

  where p_ij is the probability distribution of both the shared data D_s and the total data D_t, p_icdot and p_cdotj are the marginal probability distributions, and $H(D_s)$ and $H(D_t)$ are the entropy values for D_s and D_t, respectively.
- The Jaccard similarity index (JSI) is used to determine the threshold (CL) at which two sets of cooperative interactions (C_n) are more similar than the greatest feasible set of cooperative interactions (C_max):

$$CL = \frac{|C_n \cap C_{max}|}{|C_n \cup C_{max}|} \tag{30}$$

- Harmonic mean of packet delivery ratios (PDR) over multiple pathways is used to determine end-to-end packet delivery ($E2EPD$).

$$E2EPD = \frac{z}{\sum_{i=1}^{z} \frac{1}{PDR_i}} \tag{31}$$

  In this case, PDR_i is the packet delivery ratio for route i, and z is the total number of pathways.

The decision of trustworthy or malicious node within a network is taken based on the threshold value for which a single trust degree is utilized. To formulate a single trust value by combining the reputation component (R), with the other components of trust such as knowledge and experience. The proposed approach utilize a weighted sum method. The weighted sum method for aggregating knowledge, reputation, and experience parameters is represented by the Equation 32 whereas output the single trust value (T) denote the overall trust degree of particular IoT node.

$$T = \omega_1 * K + \omega_2 * R + \omega_3 * E \tag{32}$$

C. DEEP FEDERATED LEARNING MODEL

The proposed FedTrust approach utilizes the ANN model to train the model by modifying the federated learning concept and employing the domain, and communities to train the edge node with small parts of dataset to increase efficiency and duration of training. The complete working process of the proposed mechanism is illustrated by Algorithm 2.

The process of FedTrust begins by domain creation and allocation of global dataset D by splitting it into n parts, with each part being allocated to a dedicated server (S_i) by creating n domains (D_i). Edge node is allocated a small part of dataset by the domain server that contain high computational power. The selected edge nodes are then begins training using the provided dataset as local model training. After the creation of the local model the local model (M_{ij}) are transmitted to the corresponding domain server. Each domain server then merged these trained local model of edge node to formulate the domain trained mode (DM_i). After merging, the domain trained model is transmitted to the central server to formulate global model that is further distributed towards domain servers and shared with the edge node to provide prediction capabilities.

IV. EXPERIMENTAL SIMULATION AND ANALYSIS

We propose an ensemble learning approach for trust-based intrusion detection in IoT environment using knowledge, reputation, and experience as trust management components. Our approach is compared with two existing approaches, PoTC [18] and DDQN-Trust [20], for a comprehensive evaluation of its performance. The dataset consists of 19 trust features, and we employ an ANN as the base model. The parameters used to build the dataset were both randomly chosen and pre-established. The dataset was produced by doing the following steps:

Algorithm 2 FedTrust Federated Training Workflow

Input: Global dataset (D_{trust})
Output: Global Model (GM_{trust})

procedure Environment Creation
  Create domains: D_i (i=1, 2, …, n)
  Create dedicated servers: S_d=s=1, 2, …, n
  Domain communities: C_{ij}, (j=1, 2, …, n)
  Communities edge nodes: E_{ij}=e=1, 2, …,n
procedure Dataset Splitting & Allocation
  Split D_{trust} `into n parts:` $D_i = D/n$
  D_{trust} `allocation,` : $S_i \leftarrow D_i$
  Edge nodes selection: E_{ij}: SP_{ij}, `for training:` $E_{ij}(P_{ij} > threshold)$
  Split D_i `into chunks:` $D_i = d_{ij}$ `where` $d_{ij=D_i/m}$
  Allocate of each dataset chunk: d_{ij}, `to selected edge nodes`, E_{ij}: $E_{ij} \leftarrow d_{ij}$
procedure Training & Communities Trained Model Formulation
  Training of edge nodes: E_{ij}, `to formulate a local model,` M_{ij} : $M_{ij} = train(E_{ij}, d_{ij})$
  Local trained models transmission: M_{ij}, `to domain server,` S_i: $S_i \leftarrow M_{ij}$
  Merging of local trained models: M_{ij}, `to formulate a domain model,` DM_i: $DM_i = \text{merge}(M_{ij})$
  Transmission of domain model: DM_i, `to central server:` $CS \leftarrow DM_i$
procedure Global Model Formulation
  For each domain: i, `receive all the community trained models,` TM_j
    i. $DM_i = merge(TM_j)$
procedure Global Model Transmission
  Transmit global model: GM_{trust}, `to domain servers,` S_i: $S_i \leftarrow GM_{trust}$
  Received global model: GM_{trust}, by the central server CS
  For each community node: E_{ij}, `in the community` do
    i. Update E_{ij} `with` GM_{trust} : $E_{ij} \leftarrow GM_{trust}$
  Confirmation from edge nodes: E_{ij}, `have received updated global model,` GM_{trust}
  Exit.

- The number of nodes and edges, as well as the network's overall size, were created at random.
- Each node was given an arbitrary value based on its knowledge, reputation, and experience.
- The weights for each metric were assigned random values within the range of 0 to 1.
- All of the factors that make up the knowledge metric-how credible, accurate, reliable, compliant, capable, available, and responsive-were given arbitrary values between 0 and 1 to make up a single score.
- Parameters of nodes, such as their ratings, user engagement, and responsiveness, as well as the number of data breaches and security vulnerabilities, were given arbitrary values between 0 and 1 for the reputation measure.
- Random values between 0 and 1 were assigned to the parameters of interaction frequency, transaction success rate, transaction time, communication quality, resource utilization, data sharing behavior, cooperation level, and end-to-end packet delivery to calculate the experience metric.
- To guarantee that all parameters were between 0 and 1, the dataset was normalized using the min-max normalization method.
- Finally, the dataset was split into training and testing sets with a 70:30 ratio.

The studies may be carried out in a controlled setting with known ground truth thanks to the synthetic dataset. The dataset's diversity and ability to capture the range of practical situations was assured by randomly generating its parameters. The suggested method was assessed and compared to preexisting trust models using the synthetic dataset. The simulation results showed that the suggested method successfully identified malicious nodes in the network with high accuracy.

To optimize the model's performance, we use the Keras tuner to search for the optimal hyperparameters, including the number of hidden layers, number of neurons in each layer, activation function, optimizer, and learning rate. To simulate our proposed approach, we generate a synthetic dataset with varying numbers of nodes, ranging from 100 to 600 edge nodes, to represent an IoT network. The simulation is performed in Jupyter Notebook using Python language. We use the scikit-learn library to preprocess the dataset by cleaning and normalizing it. We then use the Keras tuner to optimize the model's hyperparameters and evaluate its performance on

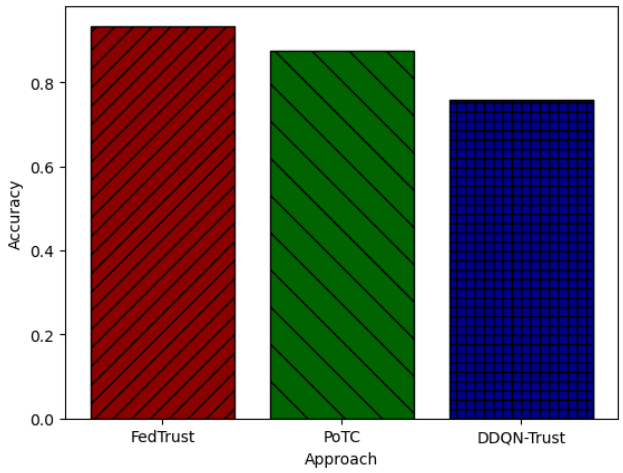


FIGURE 3. Accuracy comparison of the FedTrust with existing approaches.

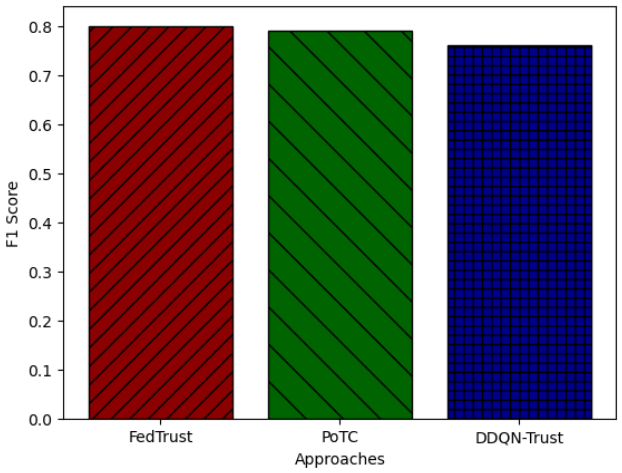


FIGURE 4. F1 score comparative analysis of the FedTrust with existing approaches.

the testing set, using accuracy, precision, recall, and F1-score as evaluation metrics.

A. ACCURACY

The evaluation of the proposed model has been performed using the accuracy metric. Accuracy is defined as the ratio of the correctly classified data points to the total number of data points in the dataset. The accuracy in the proposed model has been created as illustrated:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{33}$$

where TP (True Positive) is the number of correctly classified malicious nodes, TN (True Negative) is the number of correctly classified benign nodes, FP (False Positive) is the number of benign nodes classified as malicious, and FN (False Negative) is the number of malicious nodes classified as benign. According to the results shown in Figure 3, the proposed approach, FedTrust, outperforms the other two approaches, PoTC and DDQN-Trust, with an accuracy of 0.934. PoTC has an accuracy of 0.876, which is still a relatively good performance, but it is lower than that of the proposed approach. DDQN-Trust has an accuracy of 0.759, which is the lowest among the three approaches. FedTrust shows significant improvement in trust-based intrusion detection in the IoT context. The results suggest that the use of a privacy-enhanced trust model can significantly increase the accuracy of intrusion detection, making it a promising approach for securing IoT networks.

B. F1 SCORE

The F1 Score is a measure of a model's accuracy, which is calculated as the harmonic mean of the model's precision and recall. It is a commonly used metric to evaluate the performance of machine learning models, particularly in the context of binary classification problems. In our study, we used the F1 Score to evaluate the performance of our proposed ensemble learning approach, FedTrust, and compared it with two other existing approaches, PoTC and DDQN-Trust.

Our proposed approach, FedTrust, outperformed the other two approaches in terms of F1 Score, with a value of 0.80. This indicates that our approach has a better balance of precision and recall when compared to the other two approaches. PoTC had an F1 Score of 0.79, which is very close to our proposed approach, indicating that it is also a viable solution. DDQN-Trust had the lowest F1 Score, with a value of 0.76, indicating that it has a lower accuracy than the other two approaches. The FedTrust demonstrates its effectiveness in detecting malicious nodes in an IoT environment, as shown in Figure 4.

C. PRECISION

Precision is a performance metric that calculates the proportion of true positive instances among the total instances predicted as positive. It is computed as the ratio of true positives to the sum of true positives and false positives. Precision is an important evaluation metric, especially in scenarios where false positives can cause significant damage or false alarms. The formula for precision is as follows:

$$Precision = TP/(TP + FP) \tag{34}$$

where TP is the number of true positive instances, and FP is the number of false positive instances. The results of our simulation show that the proposed approach outperforms the existing approaches in terms of precision. The precision of the proposed approach is 0.87, while the precision of PoTC and DDQN-Trust is 0.81 and 0.74, respectively. These results indicate that the proposed approach is more effective in detecting malicious nodes with a lower false positive rate. The results of our simulation demonstrate the effectiveness of the proposed ensemble learning approach in detecting malicious nodes in an IoT environment as illustrated by Figure 5. The

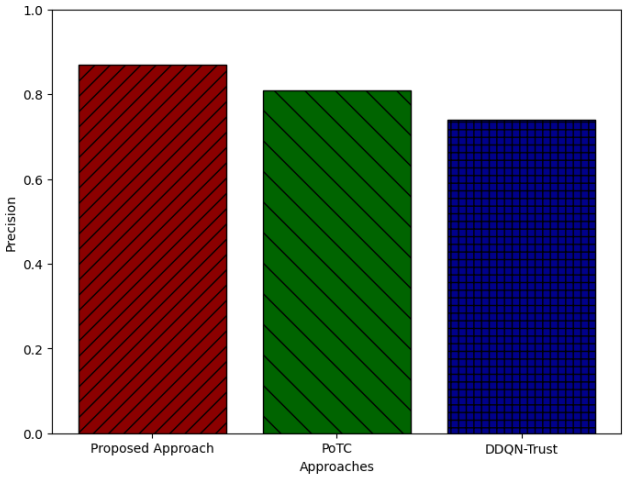


FIGURE 5. Precision comparison of the FedTrust with existing approaches.

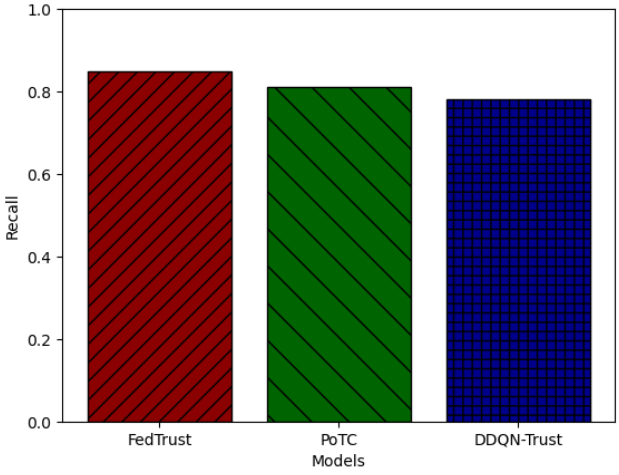


FIGURE 6. Recall comparison of the FedTrust with existing approaches.

proposed approach outperforms the existing approaches in terms of precision, indicating its potential for practical applications.

D. RECALL

Recall is a widely used performance metric in the evaluation of machine learning models, particularly in classification tasks. It is a measure of the ability of a model to identify all relevant instances of a class, which is also known as sensitivity. In the context of trust management approaches, recall is a relevant metric to evaluate the ability of a model to detect all potentially malicious entities. In this study, we evaluated the recall of three trust management approaches: FedTrust (Proposed), PoTC, and DDQN-Trust. The recall is calculated as the ratio of the true positives to the sum of true positives and false negatives, as shown in the formula below:

$$Recall = \frac{TP}{TP + FN} \tag{35}$$

where TP (True Positive) is the number of instances correctly classified as positive, and FN (False Negative) is the number of instances incorrectly classified as negative. The simulation results showed that the proposed approach achieved the highest recall value of 0.85, followed by PoTC with a recall value of 0.81, and DDQN-Trust with a recall value of 0.78 as illustrated by Figure 6.

The higher recall value of the proposed approach is indicative of its superior performance in identifying all potentially malicious entities, thus reducing the risk of allowing a malicious entity into the system. The recall values of the PoTC and DDQN-Trust approaches were close to that of the proposed approach, indicating that they are also efficient in identifying potentially malicious entities. However, the proposed approach has a higher recall value, making it more reliable in practice. The evaluation of recall performance metric in the context of trust management approaches showed that the proposed approach achieved the highest recall value, making it a more efficient approach to identify potentially malicious entities.

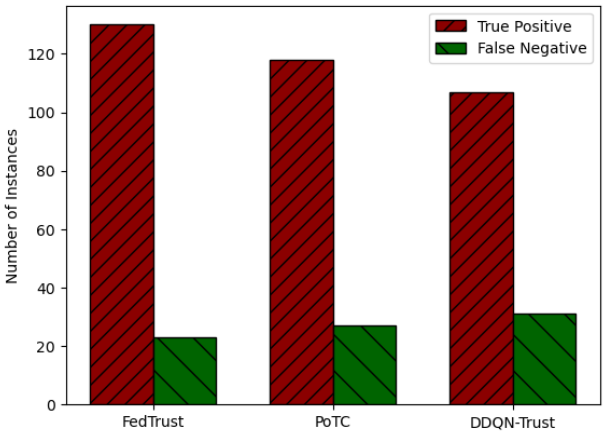


FIGURE 7. True positive and false negative values for the proposed approach and other two approaches.

E. CLASSIFICATION PERFORMANCE

The classification accuracy of the suggested method was measured by True Positive (TP) and False Negative (FN) rates. The True Positive count indicates the number of occurrences that were properly labeled as positive, whereas the False Negative count indicates the number of occurrences that were wrongly labeled as negative. Figure 7 displays the simulation results for the proposed technique, together with those for PoTC and DDQN-Trust. The figure shows that across all three simulations, the suggested method yielded the maximum number of True Positive occurrences (130, 118, and 107, respectively). To contrast, the False Negative values for the suggested method ranged from 23 in the first simulation to 27 in the second and 31 in the third.

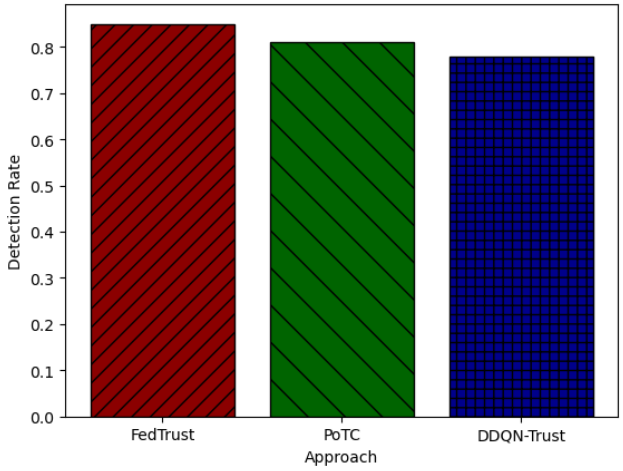


FIGURE 8. Comparison of detection rate.

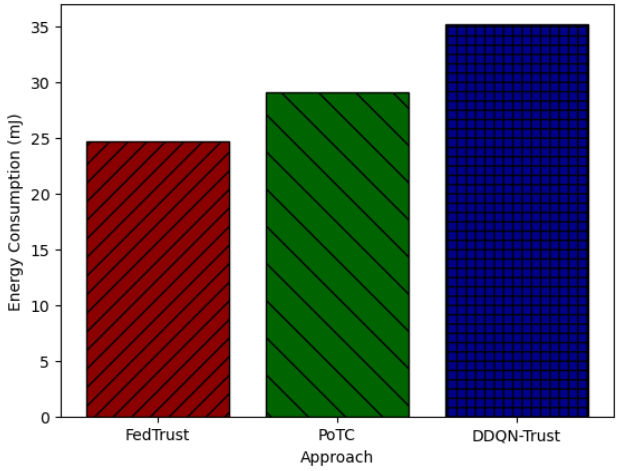


FIGURE 9. The energy consumption comparison of proposed approach with existing approaches.

When compared to the suggested method, PoTC and DDQN-Trust both had lower True Positive values and greater False Negative values. This suggests that the suggested method is more adept at accurately categorizing positive examples while reducing the number of wrongly categorized negative examples. The success of a classification model depends on a number of variables, including the size and quality of the dataset, the features used, and the classification technique used. In this research, we show that our suggested method outperforms the other two methods in terms of True Positive and False Negative rates of classification.

F. DETECTION RATE

The detection rate is a key performance indicator for how well a technique can spot malicious nodes. Here, we show the detection rate that FedTrust was able to accomplish and compare it to the detection rates of PoTC and DDQN-Trust, two competing methods. In Figure 8, based on the simulation results, it was clear that the suggested FedTrust method outperformed the other two in terms of detection rate. The suggested method showed a higher detection rate than both PoTC and DDQN-Trust (0.85 vs. 0.81 and 0.78, respectively).

More accurately detecting malicious nodes with fewer false positives is what we mean by a high detection rate. Out of 130 cases tested, our suggested method successfully categorized 23 as false negatives (FN) and true positives (TP). That our method successfully identifies malicious nodes while reducing false positives is a strong indicator of its efficacy. The detection rates of the PoTC and DDQN-Trust techniques were found to be lower in the comparison study. A detection rate of 0.81 indicates that out of a total of 200 cases, PoTC properly recognized 118 as TP and wrongly categorized 27 as FN. Among the three methods, DDQN-Trust's 0.78 detection rate was the lowest. Using this method, we were able to accurately identify 107 cases as TP and mistakenly label 31 cases as FN.

G. ENERGY CONSUMPTION

Trust and reputation management strategies for IoT systems must take energy consumption into account throughout the planning and implementation stages. Here, we compare the FedTrust strategy's energy use to that of the PoTC and DDQN-Trust strategies, and draw conclusions about which one is more efficient. The simulation results (Figure 9) shown that our method, with a value of 24.7 mJ, used the least amount of energy. The lowest energy usage was achieved by PoTC (29.1 mJ), while the highest was achieved by DDQN-Trust (35.2 mJ).

Our suggested method uses less power since it makes better use of available resources and employs optimized algorithms. Our method is more efficient than the other two in terms of energy usage and resource consumption since it relies less on raw accuracy and detection rate. The energy consumption figures for the PoTC and DDQN-Trust techniques were found to be higher in the comparison study. The 29.1 mJ of energy required by PoTC is more than our recommended method. When compared to the other two methods, DDQN-Trust is the least energy-efficient option, with a value of 35.2 mJ for its energy usage.

V. CONCLUSION

A novel ensemble learning approach for detecting malicious nodes in an IoT environment using trust management components such as knowledge, reputation, and experience. The proposed approach utilizes an ANN as a base model to classify the nodes into malicious or benign. To optimize the performance of the model, we will use the Keras tuner to search for the optimal hyperparameters of the ANN, such as the number of hidden layers, number of neurons in each layer, activation function, optimizer, and learning rate. The proposed architecture consists of three main components: the data acquisition module, the trust management module, and the decision-making module. The data acquisition module

collects data from the IoT devices, which is then preprocessed and fed into the trust management module. The trust management module utilizes the ensemble learning approach to identify malicious nodes based on their behavior, reputation, and experience. The decision-making module takes the output of the trust management module and decides on the action to be taken, such as isolating the malicious nodes or increasing their security level. The proposed approach can be further extended to investigate the scalability and robustness of the proposed approach in real-world scenarios with a large number of nodes and complex IoT architectures.

REFERENCES

[1] Y. Li, S. Xie, Z. Wan, H. Lv, H. Song, and Z. Lv, "Graph-powered learning methods in the Internet of Things: A survey," *Mach. Learn. Appl.*, vol. 11, Mar. 2023, Art. no. 100441.

[2] K. Ramezanpour, J. Jagannath, and A. Jagannath, "Security and privacy vulnerabilities of 5G/6G and WiFi 6: Survey and research directions from a coexistence perspective," *Comput. Netw.*, vol. 221, Feb. 2023, Art. no. 109515.

[3] Y. Liu, J. Wang, Z. Yan, Z. Wan, and R. Jäntti, "A survey on blockchain-based trust management for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5898–5922, Apr. 2023.

[4] J. A. Josephine, S. Senthilkumar, R. Rajkumar, and A. Kumar, "Detection of authorized nodes to provide an optimal secure communication in amalgamated internet MANET," in *Proc. Int. Conf. Internet Things*. Berlin, Germany: Springer, 2023, pp. 93–102.

[5] A. Farraj, "Coordinated security measures for industrial IoT against eavesdropping," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2023, pp. 1–5.

[6] P. D. Rosero-Montalvo, Z. István, P. Tözün, and W. Hernandez, "Hybrid anomaly detection model on trusted IoT devices," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10959–10969, Jun. 2023.

[7] P. Benlloch-Caballero, Q. Wang, and J. M. A. Calero, "Distributed dual-layer autonomous closed loops for self-protection of 5G/6G IoT networks from distributed denial of service attacks," *Comput. Netw.*, vol. 222, Feb. 2023, Art. no. 109526.

[8] M. Abbasi, M. Plaza-Hernández, and Y. Mezquita, "Security of IoT application layer: Requirements, threats, and solutions," in *Proc. 13th Int. Symp. Ambient Intell.* Berlin, Germany: Springer, 2023, pp. 86–100.

[9] R. Verma and S. Chandra, "RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu," *Eng. Appl. Artif. Intell.*, vol. 118, Feb. 2023, Art. no. 105670.

[10] S. Subramani, M. Selvi, A. Kannan, and S. K. Svn, "Review of security methods based on classical cryptography and quantum cryptography," *Cybern. Syst.*, pp. 1–19, Jan. 2023.

[11] L. Ouyang, F. Wang, Y. Tian, X. Jia, H. Qi, and G. Wang, "Artificial identification: A novel privacy framework for federated learning based on blockchain," *IEEE Trans. Computat. Social Syst.*, early access, Feb. 1, 2023, doi: 10.1109/TCSS.2022.3226861.

[12] A. O. Philip and R. K. Saravanaguru, "Multisource traffic incident reporting and evidence management in Internet of Vehicles using machine learning and blockchain," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105630.

[13] P. K. Laboso, A. Martin, and P. Thiyagarajan, "Blockchain technologies in data science: Challenges and benefits," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Mar. 2023, pp. 1323–1328.

[14] Y. Lin, H. Du, D. Niyato, J. Nie, J. Zhang, Y. Cheng, and Z. Yang, "Blockchain-aided secure semantic communication for AI-generated content in metaverse," *IEEE Open J. Comput. Soc.*, vol. 4, pp. 72–83, 2023.

[15] M. Westerkamp and A. Küpper, "Instant function calls using synchronized cross-blockchain smart contracts," *IEEE Trans. Netw. Service Manage.*, early access, Jan. 12, 2023, doi: 10.1109/TNSM.2023.3236437.

[16] K. Ashok and S. Gopikrishnan, "Statistical analysis of remote health monitoring based IoT security models & deployments from a pragmatic perspective," *IEEE Access*, vol. 11, pp. 2621–2651, 2023.

[17] A. George and A. Ravindran, "Scalable approximate computing techniques for latency and bandwidth constrained IoT edge," in *Science and Technologies for Smart Cities*. Amsterdam, The Netherlands: Springer, Dec. 2021, pp. 274–292.

[18] L. Bi, T. Muazu, and O. Samuel, "IoT: A decentralized trust management system using blockchain-empowered federated learning," *Sustainability*, vol. 15, no. 1, p. 374, Dec. 2022.

[19] S. Kalantar, M. Jafari, and M. Hashemipour, "Energy and load balancing routing protocol for IoT," *Int. J. Commun. Syst.*, vol. 36, no. 2, Jan. 2023, Art. no. e5371.

[20] G. Rjoub, O. A. Wahab, J. Bentahar, and A. Bataineh, "Trust-driven reinforcement selection strategy for federated learning on IoT devices," *Computing*, pp. 1–23, Apr. 2022.

[21] N. Gholizadeh, N. Kazemi, and P. Musilek, "A comparative study of reinforcement learning algorithms for distribution network reconfiguration with deep *Q*-learning-based action sampling," *IEEE Access*, vol. 11, pp. 13714–13723, 2023.

[22] M. Mendieta, C. Neff, D. Lingerfelt, C. Beam, A. George, S. Rogers, A. Ravindran, and H. Tabkhi, "A novel application/infrastructure co-design approach for real-time edge video analytics," in *Proc. SoutheastCon*, Apr. 2019, pp. 1–7.

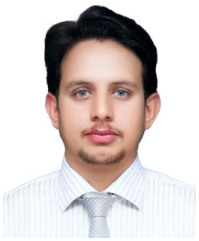
[23] G. Rjoub, O. A. Wahab, J. Bentahar, R. Cohen, and A. S. Bataineh, "Trust-augmented deep reinforcement learning for federated learning client selection," *Inf. Syst. Frontiers*, pp. 1–18, Jul. 2022.

[24] O. B. Adedoyin and E. Soykan, "COVID-19 pandemic and online learning: The challenges and opportunities," *Interact. Learn. Environ.*, vol. 31, no. 2, pp. 863–875, Feb. 2023.

[25] M. Sarhan, W. W. Lo, S. Layeghy, and M. Portmann, "HBFL: A hierarchical blockchain-based federated learning framework for collaborative IoT intrusion detection," *Comput. Electr. Eng.*, vol. 103, Oct. 2022, Art. no. 108379.

[26] X. Hei, X. Yin, Y. Wang, J. Ren, and L. Zhu, "A trusted feature aggregator federated learning for distributed malicious attack detection," *Comput. Secur.*, vol. 99, Dec. 2020, Art. no. 102033.

[27] S. Otoum, I. A. Ridhawi, and H. Mouftah, "Securing critical IoT infrastructures with blockchain-supported federated learning," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2592–2601, Feb. 2022.

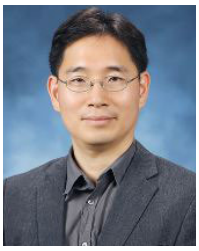


KAMRAN AHMAD AWAN received the B.S. and M.S. degrees in computer science from the Department of Information Technology, The University of Haripur, Pakistan, in 2015 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include trust management in the Internet of Things, blockchain, security in metaverse, and information security.

IKRAM UD DIN (Senior Member, IEEE) received the Ph.D. degree in computer science from the School of Computing, Universiti Utara Malaysia (UUM), in 2016. He was the IEEE UUM Student Branch Professional Chair. He is currently an Associate Professor with the Department of Information Technology, The University of Haripur. He has 12 years of teaching and research experience in different universities/organizations. His current research interests include traffic measurement and analysis for monitoring quality of service, mobility and cache management in information-centric networking, metaverse, and the Internet of Things.

MAHDI ZAREEI (Senior Member, IEEE) received the M.Sc. degree in computer networks from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Malaysia-Japan International Institute of Technology, University of Technology, Malaysia, in 2016. In 2017, he joined Tecnologico de Monterrey's School of Engineering and Sciences as a Postdoctoral Fellow. He ia a advancing to Research Professor, in 2019. His research interests include wireless sensor and ad hoc networks, information security, and the applied machine learning and natural language processing. He is a distinguished researcher and professor. He is a Level I member of the Mexican National Researchers System and serves as an Associate Editor for IEEE Access, *PLOS One, and Ad Hoc and Sensor Wireless Networks Journals*.

AHMAD ALMOGREN (Senior Member, IEEE) received the Ph.D. degree in computer science from Southern Methodist University, Dallas, TX, USA, in 2002. He is currently a Professor with the Department of Computer Science, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia, where he is also the Director of the Cyber Security Chair, CCIS. Previously, he was the Vice Dean of the Development and Quality with CCIS. He was also the Dean of the College of Computer and Information Sciences and the Head of the Academic Accreditation Council, Al Yamamah University. His research interests include mobile-pervasive computing and cyber security. He served as the General Chair for the IEEE Smart World Symposium and a Technical Program Committee member for numerous international conferences/workshops, such as IEEE CCNC, ACM BodyNets, and IEEE HPCC.



BYUNG SEO-KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. From 1997 to 1999, he was Motorola Korea Ltd., Paju, South Korea, as a Computer Integrated Manufacturing (CIM) Engineer in advanced technology research and development (ATR&D). From January 2005 to August 2007, he was with Motorola Inc., Schaumburg, IL, USA, as a Senior Software Engineer in networks and enterprises. His research focus at Motorola Inc. were designing protocol and network architecture of wireless broadband mission critical communications. From 2012 to 2014, he was the Chairman with the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His work has appeared in around 174 publications and 25 patents. His research interests include the design and development of efficient wireless/wired networks, including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He served as a member of Sejong-City Construction Review Committee and Ansan-City Design Advisory Board and a TPC Member for the IEEE VTC 2014-Spring and the EAI FUTURE2016, and ICGHIT 2016 and 2019 conferences. He was also served as the General Chair for 3rd IWWCN 2017 and the Guest Editor for Special Issues of *International Journal of Distributed Sensor Networks* (SAGE), IEEE Access, *Sensors* (MDPI), and *Journal of the Institute of Electronics and Information Engineers*. He is an Associate Editor of IEEE Access.

JESÚS ARTURO PÉREZ-DÍAZ (Member, IEEE) received the B.Sc. degree in computer science from the Autonomous University of Aguascalientes, in 1995, and the Ph.D. degree in new advances in computer science systems from Universidad de Oviedo, in 2000. From 2000 to 2002, he became a Full Associate Professor with the University of Oviedo. He is currently a Researcher and a Professor with Tecnológico de Monterrey—Campus Guadalajara, Mexico, and a member of the Mexican Researchers National System. His research interests include cyber security in SDN and the design of communications protocols, where he has supervised several master's and Ph.D. theses in the field. He was recognized by the COIMBRA Group as one of the Best Young Latin-American Researchers, in 2006, and received a research stay with Louvain Le Nouveau University, Belgium. He has been awarded by the CIGRE and Intel for the development of innovative systems. He was a recipient of the Best Student Award for his B.Sc. degree from the Autonomous University of Aguascalientes.