International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Sep-2023)

http://dx.doi.org/10.12785/ijcds/140170

Trust Management in Social Internet of Things: Challenges and Future Directions

Santhosh Kumari[1], S M Dilip Kumar[2] and Venugopal K R[3]

[1,2,3]*Department of Computer Science and Engineering, UVCE, Bengaluru, India*

Received 28 Dec. 2022, Revised 06 Aug. 2023, Accepted 10 Aug. 2023, Published 01 Sep. 2023

Abstract: IoT is expanded by leveraging social networking ideas to build a social network of interlinked smart devices called Social Objects (SO), and the resulting network is called the Social Internet of Things (SIoT). These SOs have social features that allow them to find other SOs in their environment and establish social interactions with them. Trust Management Systems (TMS) comprehends how the data supplied by the communicating parties must be processed based on the object's behavior in order to establish reliable autonomous communications. The literature on TMS in SIoT is limited, and the existing paper's review of issues, challenges, and future directions is not complete. This paper first presents the trust management concepts in SIoT. Second, the existing TMS for SIoT proposed in papers throughout the previous seven years (2017–2023) are categorized as process-based TMS, context-based TMS, blockchain-based TMS, and edge-based TMS. These models are analyzed in terms of trust features, aggregation techniques, trust update mechanisms, trust propagation strategies, evaluation tools, and performance metrics. The percentage of effort exhibited by various TMS in solving issues in SIoT are: residual energy of a node (14%) and scalability (19%) are given less emphasis, while most of the TMS have focused on resiliency against BMA (81%) and BSA (71%). Third, the paper discusses research challenges and future directions investigated by the survey that help the researchers develop a robust, adaptable and resilient TMS.

Keywords: Blockchain-based TMS, Context-aware TMS, Edge-based TMS, Internet of Things, Process based TMS, Social Internet of Things, Social Objects, Trust Management System.

I. Introduction

Internet of Things (IoT) is a collection of networked devices that communicate and share data wirelessly without the need for human interaction. Smart objects in IoT are described as autonomous, independent devices with the capacity to detect their environment and analyse the data gathered from them and other smart objects nearby [1]. The incorporation of social networking fundamentals into IoT have paved way to new paradigm called Social Internet of Things (SIoT), where SOs can lay out friendly connections in an independent manner as for their proprietors [2]. The SIoT ecosystem is made up of a variety of SOs with distinct behaviors, and they can identify required services by leveraging their social relationships with nearby nodes if there is sufficient trust between them. Malevolent objects, also known as misbehaving objects, are nodes that are under the influence of an adversary and target other objects to boost their own profits while shutting off the services of others. These nodes might also damage the reputation of nodes with honest behaviour or enhance the credibility of nodes with malevolent intent, impairing the network's fundamental functionality. To build a more secure and promising SIoT environment, malicious nodes must be banned and trust must be established between communicating nodes. The SIoT objects can establish a level of trustworthiness by leveraging the degree of interaction among friendly objects. As the degree of trust between objects grows, the nodes cooperate and interact with their trusted neighboring nodes by providing their services, which eventually limits their vulnerability to malevolent nodes.

In this situation, trust management is essential as it assists the SIoT objects get through perceptions of uncertainty and danger brought on by exposure to malevolent nodes. To mitigate the effect of malicious device's abnormal functionality, the TMS encourage objects to work honestly and productively by identifying the most trustworthy trustee for each trustor.

A. Motivation

Numerous TMS and trust models were studied by researchers for the IoT environment [9], [10], [11], yet they cannot be effectively utilised in SIoT environment because they do not consider social factors, relationships between objects, or social trust qualities. Only a few research studies

E-mail address: santhoshkumariy@gmail.com, dilipkumarsm.uvce@gmail.com, venugopalkr@gmail.com

TABLE I. Summary on related survey on Trust Management Models in SIoT

Ref#	Contribution	TMP	ITM	TMM	RC	FD
[3], 2019	Recent research papers on the SIoT environment's service composition, service discovery, relationship management and trust management are reviewed.	✗	✗	✔	✔	✔
[4], 2019	The SIoT trust management presents an overview of previous SIoT trust management studies and compares them to various performance metrics and trust-related attacks.	✗	✗	✔	✗	✗
[5], 2019	It gives a complete comparison of protocols and architectures between IoT and SIoT. It also classifies and compares several trust management models based on the trust management process for SIoT.	✔✔	✔	✔	✔	✗
[6], 2020	Presented the fundamental concepts of SIoT and trust management, SIoT Architectures comparision, trust management systems are categorised and also discussed about open research challenges	✔✔	✗	✔	✔	✔
[7], 2020	Reviewed the architecture of SIoT, key features, parameters and challenges of SIoT components such as Trust Management, Relationship Management, web services and information processing.	✗	✗	✔	✔	✗
[8], 2022	Presented the concepts of trust management, classified TMS into four categories and thier strength and limitations analysed	✔✔	✗	✔	✔	✔
our paper, 2023	Presents fundamentals of SIoT and trust management, classifies TMS as process-based, context based, blockchain based and edge-based systems. It also discussed TMS's strengths and limitations, research challenges and future directions	✔✔	✔✔	✔✔	✔✔	✔✔

TMP: Trust Management Process; ITM: Issues in Trust Management; TMM: Trust Management Models; RC: Research Challenges; FD: Future Directions; ✔✔: Fully Addressed; ✔: Partial; ✗: Unaddressed.

on trust management in SIoT are presently available in the scientific literature, as shown in Table I. The authors of [3] discussed the fundamentals of SIoT and indicated key topics for further research, including service composition, relationship management, service discovery, and trust management. It also reviewed the most recent articles of thrust area. However, their thorough analysis does not compare the latest trust management schemes suggested for the SIoT. Paper [4] reviews TMS for SIoT and compares the models' adaptability, power efficiency, robustness, scalability, and survival. But this survey lacks reviewing trust management process, issues and open research challenges. The authors in [5] have compared various trust management solution according to the process of trust management, trust functions, defense against attacks and different environment(wireless sensor network, online social network, IoT, SIoT). This paper lacks detailed survey of TMS for SIoT and clarification of challenges in developing a more secure trust models for SIoT. The authors in [6] have clarified the difference between IoT and SIoT, comparision of SIoT architecture, comparative analysis of TMS and open research challenges for SIoT. This paper lacks detailed comparision and analysis of TMS (blockchain), issues and open research challenges. The authors in [7] have reviewed the architecture and publications on information processing, web services, relationship management and trust management. Further, these components were evaluated in terms of scalability, navigability, accuracy, resiliency, time etc., yet this paper lacks detailed review of trust management process, issues, research challenges and future directions. The authors in [8] provide a review on components of trust management, constrasted various trust management models, a summary of trust in

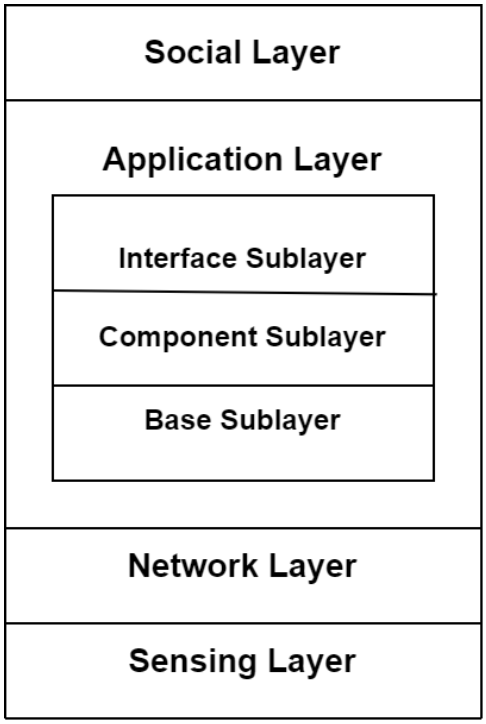

Figure 1. SIoT Architecture.

platforms and applications based on the SIoT, and future research directions on trust management in SIoT. But it lacks comprehensive research challenges and future directions. To provide researchers and developers with a clear picture of the cutting-edge concepts and their shortcomings, this study provides a detailed review of the TMS proposed for SIoT during the previous seven years. The following are the main contributions:

1) Extensive survey on process-based, context-based, blockchain-based and edge-based TMS is provided.

2) It is shown how existing schemes have been analysed in terms of their trust features, aggregation techniques, trust update mechanisms, trust propagation strategy, evaluation tools, and performance metrics.

3) Strengths and limitations, open research challenges and research directions are identified by a thorough analysis of the schemes discussed.

B. SIoT Architecture

The SIoT architecture is depicted in Figure 1. The four layers are the sensor, network, application, and social layers. Below is a description of these sublayers' elements and functions [5].

1) *Sensing Layer*: This layer deals with sensing and actuation. This layer's primary function is to gather environmental data by utilizing various real-world and virtual things [12].

2) *Network Layer*: This layer transmits the pre-processed information gathered by IoT devices to the application layer for information processing and vice versa via the communication channel.

3) *Application Layer*: The interface sublayer, component sublayer, and base sublayer are the three sublayers that make up the application layer.

   a) *Base Sublayer:* It has a database that keeps information about object owners, their profiles, their social connections, and the tasks that the objects carry out.

   b) *Component Sublayer*: It aids in implementing the basic functionalities of SIoT system such as ID Management, Profiling, Relationship Management (RM), Service Discovery (SD), Owner Control (OC), Service Composition (SC) and Trust Management (TM) modules. *ID Management* module assigns IDs to all objects and keeps them up to date. *Profiling* module is responsible for configuring object information. *OC* module defines the kind of relationships that objects can have, the type of information they are willing to share, the kind of entities that can access the information and various interactions these objects can carry out. *RM* module incorporates the skill of creating and sustaining relationships. It enables objects to start, keep updating, and terminate relationships according to the policies specified in the OC module. *SD* module helps the objects to locate the services they require from other objects. *SC* module aids in providing the services discovered using the SD module by the objects. *TM* module comprehends how the data supplied by the communicating parties is processed and by considering the object's behaviour reliability is established [13].

   c) *Interface Sublayer*: This layer provides a medium for facilitating interactions between diverse entities. It includes interfaces for people, their objects, and their services.

4) *Social Layer:* This layer facilitates social interactions between diverse objects by using the data that it receives from the application layer's components.

C. Applications of SIoT

Some of the applications of SIoT are: health sector, smart cities and homes, industrial sector, traffic management and agriculture. In health sector it helps in overseeing healthcare systems [14], [15]. In Smart cities it helps in managing cities and running smart homes, monitoring the environment, eas-

ing remote monitoring and controlling facilities like elevators [16], [17] . In industrial sectors, it helps in administering industrial plant and operating vending machines segment [18]. In traffic management system, it assists in tracking vehicles and provides route information [19], [20]. It is used in agriculture for monitoring soil moisture, temperature, and so on. [21], [22].

II. Background

Trust is one of the most important aspect of human life, for people to form connections with one another. The belief of one person (the trustor) in another person (the trustee) is the definition of trust in its simplest form [23]. Since the SOs in SIoT paradigm mimics human intrinsic nature, these SOs assess the trust parameter of other SOs before starting a communication and if the level of trust is satisfied, then they exchange their services. Therefore trust is the key component of SIoT. Trust also increases the security and privacy of data and enhances customer decision to use SIoT technology [24].

Subjective trust and objective trust are two major categories of trust. From a social perspective, **subjective trust** is the process through which each node determines the trustworthiness of its friends based on personal experience (direct trust). If nodes N_i and N_j are not friends, then a chain of friendships is employed to determine who is trustworthy (opinion of friends or feedback) [2]. Some works that used subjective trust to determine a node's trust score are: In *DTrustInfer* [25] a node's trust score is assessed using honesty, community of interest, cooperativeness and energy status of a device. In CBSTM-IoT [26], trust score is composed of node's computation power, context importance, confidence, feedback, centrality, friendship and SIoT relationship. The context-aware trust model in [27] computes the trustworthiness score of a node by aggregating direct trust (historical interactions), recommendations, centrality and community interest.

In peer to peer settings, where data about each node is disseminated and saved using a DHT (Distributed Hash Table) structure. Every node can see this data, but only specialised nodes known as Pre-Trusted Objects (PTOs) have control over it. The trustworthiness of node pj as observed by the whole network is called **Objective trust** [2]. Some examples of trust models that gather objective trust are: the trust model in [28] deploys a reputation server to calculate the nodes reputation score, which inturn is used as a trustworthiness score of a node. When each transaction is finished, the SR sends feedback to the reputation server about the received service's quality and its social relationship with the SP. In [29], a node's trustworthiness is defined by node's centrality and similarity metrics. In MAG-SIoT model, a node's trust score is calculated based on a predetermined set of social relationships that each SIoT member has [30].

A. Trust Management Process in SIoT

The trust management process is done in four phases. They are *information gathering, trust calculation, trust decision, tust update and reward/punish* as discussed below:

1) *Information Gathering*: The SOs gather data about the entities from which they request services or to which they deliver services by tracking trust metrics such as *Quality of Service (QoS) trust* and *social trust*. QoS trust describes the capability of a device to execute a task requested by a node. It is measured in the form of competency, co-operativeness, reliability, capability to complete tasks, packet delivery ratio, energy usage, and end-to-end packet forwarding ratio. Social trust is a device's willingness and commitment to execute a service request. It is assessed using intimacy, benevolence, privacy, centrality, friendship, connectedness, social contact, community of interest and unselfishness [31], [32].

2) *Trust Aggregation*: Nodes willing to offer a specific service is determined by the TMS and its trust parameters gathered in earlier stage are aggregated to a single value using any of these functions such as weighted sum, Bayesian inference, belief theory, fuzzy logic, regression analysis and Machine Learning (ML) algorithms [33][34]. This single value obtained can be a binary value or a numeric value used to decide whether a node is malevolent or benevolent and also to rank services of a node.

3) *Trust Decision*: After calulating the trustworthiness value of SP, this value is used to decide whether SP is malicious or not. In making this choice threshold-based or context-based decision is used. In threshold-based decision method, choice is made by either implementing a rank-based function or a threshold value. Context-based decision method creates policies utilising contextual information (location, time, energy status), to classify a node as malicious or not.[8].

4) *Trust Propagation*: Nodes that need service communicates with the nodes that provides service and observes the quality of services it receives. Once the requested transaction is complete, the requesting node updates the provider's trust values depending on its QoS. There are two techniques to update trust: *event-driven* and *time-driven*. A node's trust score is updated in an event-driven trust update once an event or transaction has taken place. On regular basis trust value is updated in line with predefined time intervals in time driven trust

updating.

5) *Reward or Punish*: The service requestor (SR) determines whether the service provider(SP) node should be rewarded or penalized based on the actual transactions, depending on how well the service was provided.

B. Context Life Cycle in Context-based Trust Management

The term "context" refers to any type of information that can describe the status of an entity or the features of a particular environment in which an object interacts with an application and with other entities. The format, size, and representation of the context information might all differ. The ability of a system, applications, services, or actuator to adapt to a particular situation is known as context awareness. Presentation, execution, and tagging are the characteristics of context-aware systems are emphasised in several research papers [35], [36].

The period between the acquisition of a context and its spread is known as the context life. The context life cycle typically contains four phases: Obtaining context comes first, followed by modelling context, reasoning context, and distributing context [37]. In the initial stage, context is gathered from a number of sources that include, virtual and actual sensors, and is formatted in a variety of ways. The second phase involves using modelling approaches to transform the gathered context data into a unified format that can be read and processed by machines, allowing it to be shared and understood. In the third step, initially, the data is cleaned by filling in gaps, investigating context discrepancies, removing outliers, and employing data mining algorithms. Furthermore, for a better understanding, the context value is derived by mapping the context that is available from sensors to context sets derived from a high-level context. Finally, The fourth stage involves providing customers or appropriate apps with pertinent context information. The final context information is either supplied to users or utilize it locally to make decisions.

III. Related Works

This section explores the different kinds of trust management schemes namely process-based TMS, context-aware TMS, blockchain-based TMS and edge-based TMS as depicted in Figure 2.

A. Process-based TMS

This section classifies and examines several trust models based on different techniques used during various phases of trust management process, as depicted in Table II.

1) *Trust Source*: In SIoT environment, trust information gathered through direct communication between SOs is called *direct trust* [52]. In the absence of direct trust between the communicating SOs then trust information is collected from their common friends or through chain of friendship called *indirect trust*. Other authors refer to indirect trust in other ways, using terms like reputation, recommendation, rating, and feedback [53].

a) *Direct Trust (Knowledge)*: The trustworthiness of objects is assessed by the authors in [42] using the direct trust, centrality, community interest, cooperativeness, and service score trust criteria. An object's trust values are updated on a regular basis. An object receives a service score if it provides the desired services; otherwise, it receives a penalty. A node has increased odds of being malicious the more times it is penalised. By taking into account the past behaviour of a node, its future behaviour is forecasted as malicious node or benign node. This aids in shielding against selective forwarding attacks. The TMS was simulated utilising Network Simulator-3 and SoCNetV1.9 a social network visualizer tool.

b) *Indirect Trust*: The authors in [28] proposed a trust system called *Guarantor and Reputation model*. This model employs a reputation server to determine a node's reputation score, which is used to evaluate trustworthiness of a node. After completion of every transaction the SR sends feedback to the reputation server about the services it has received as good/bad along with its social relationship of the SP. Further, the requestor node gives credits to the SP if the service was satisfactory otherwise it forfeits the node. A nodes reputation is determined according to its feedback and by providing various relationships with different weights. Guarantees for an object's behaviour are provided by the credits and forfeit rates. In addition, every good service increases its reputation score by 0.1 times and if it defaults its score is reduced by 0.3 times. If the reputation score falls below a certain threshold, the nodes cannot participate in further communications and thereby isolating malicious nodes.

c) *Hybrid*: The authors in [54] have developed a detailed trust model known as Reputation, Experience and Knowledge (REK). The trust metrics used are recommendations (third party opinion), experience and first hand observations (direct trust). When carrying out a task, participants in a transaction offer good and negative feedback after completing a task. This feedback is then compiled and used as a node's reputation. The trust attributes—current relationship status, interaction frequency, and interaction values—uncooperativeness, cooperation, and

TABLE II. Trust Management Process based Classification of TMS

Ref#	Trust Source	Trust Features	Aggregation Techniques	Trust update	Propagation
[38]	Direct and Indirect trust	Honesty, Cooperativeness, Community Interest, Recommendations	Weighted Sum	Event driven	Distributed
[28]	Indirect trust	Reputation, Social relationships	Weighted Sum	Event driven	Centralized
[39]	Indirect trust	Recommendation, Reputation	Weighted Sum	NA	Hybrid
[40]	Direct and Indirect trust	Recommendations, Knowledge, Reputation	Fuzzy logic based	Event driven	Distributed
[41]	Direct and Indirect trust	Trust level capability, Sociability	Weighted sum	Time driven	Centralized
[42]	Direct trust	Cooperativeness, Centrality, Service Score, Community interest	Bayesian	Time driven	NA
[25]	Direct and Indirect trust	Honesty, Centrality Cooperativeness, Dependability, Community Interest, Energy, Recomendation	Weighted Sum	Time driven	Centralized
[43]	Direct and Indirect trust	Recommendations, Vehicles' Location Related Honesty and Honesty Human Factor (HHF)	Weighted Sum	Event driven	Centralized (Trusted Authority)
[44]	Direct and Indirect trust	Relation-strength based trust and Similarity based trust	Weighted Sum	Event driven	Distributed
[45]	Direct and Indirect trust	Computational capabilities, Relationship factor, External opinion and Dynamic Knowledge	Weighted Sum	Event driven	Distributed
[46]	Indirect trust	Reputation	Weighted sum	Event driven	Hybrid
[29]	Direct trust	Similarity and Centrality	Weighted Sum	Event driven	Distributed
[34]	Direct and Indirect trust	Dynamically selected	Artificial Neural Network (ANN) algorithm	Time driven	Distributed
[47]	Direct and Indirect trust	Credibility, Reputation, Direct experience, Rating Frequency, Rating trend, Similarity, Relationship strength, Fluctuation, Device trust, Service trust	ML and Deep Learning methods	Event driven	Hybrid
[48]	Direct and Indirect trust	Similarity in friendship, Community of Interest, Cooperativeness	K-means Algorithm	NA	Distributed
[49]	Direct	Storing capabilities, Competence, Co-operativeness, Honesty	Weighted Sum, Reccurant Neural Network	Event Driven	Hybrid.
[50]	Direct and Indirect trust	Direct trust score, Reliability, Benovelence, Credibility, Recommendations, Degree of relationship	ANN model	Event driven	Distributed
[51]	Direct	Availability, Credibility, Honesty	Weighted Sum, SVM, ANN	Event driven	Centralized

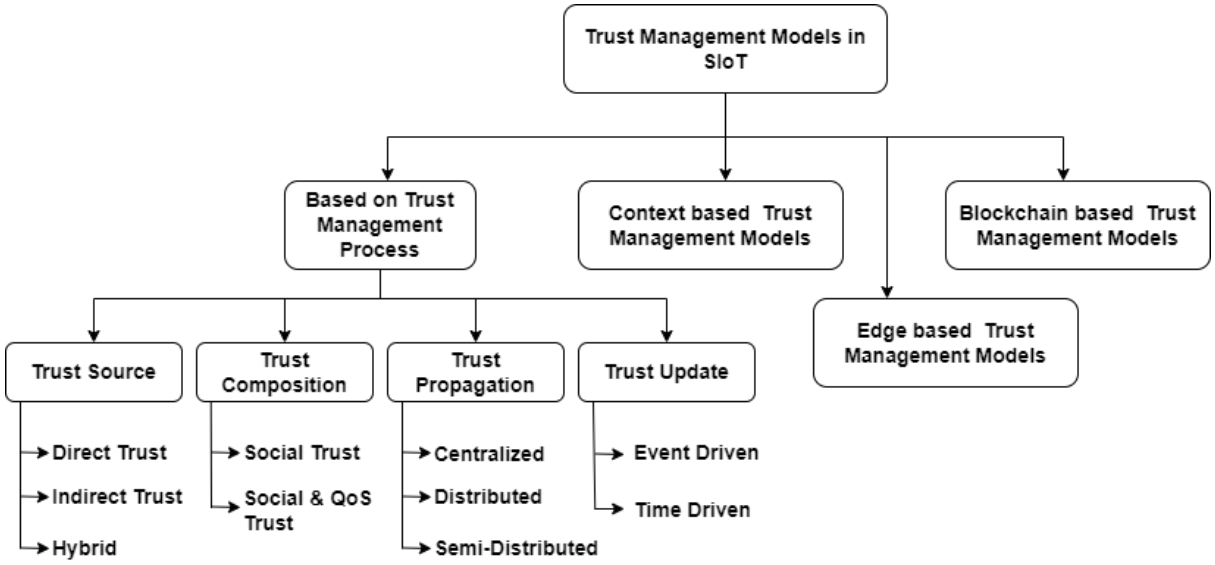

Figure 2. Classification of Trust Management Models.

neutrality—are used to calculate experience. Three different types of attributes—ability, goodness, and integrity—are used to quantify knowledge as a direct trust.

2) *Trust Composition*: Trust is composed of QoS trust and/or social trust features.

   a) *Social Trust:* The authors in [39] have presented a trust and reputation model (TRM-SIoT). This model calculates the trust score of a node using direct observations and by interacting with friends of the node whose trust has to be estimated, or by using the COSMOS platform. Platform and neighbourhood methods are used to determine reputation. The neighbours or social circle are used to obtain the reputation indices. The platform offers a reputation index when the social circle fails to provide a node's feedback. The above model is an illustration of a hybrid method since reputation is measured both centrally through a COSMOS platform and feedback from friends are gathered in a decentralized manner.

   b) *Social and QoS Trust:* The authors in [55] have presented a trust model. Trust is calculated by considering reccomendations of a node, social relationships, centrality and energy level of a device. This model makes use of social metrics (social relationships, reccomendations) and QoS metrics (centrality, energy level). Experiments in simulation show the system's efficacy in rating accuracy, dynamic behaviour, and network stability.

3) *Trust Propagation:* A centralised, distributed or semi-distributed methods are used to propagate trust in the system.

   a) *Centralised*: TMS based on communities of interest is proposed by the authors of [41]. In this model, the network is shown as set of node-clusters where each cluster is comprised of set of nodes with similar interest (Community of Interest). Every cluster has a cluster head called *administrator*. It is also incharge of calculating and keeping track of the trust score of the cluster members and to expel a misbehaving entity from the cluster and it is blacklisted on SIoT server. Each node wishing to join the SIoT network is registered and authenticated by the SIoT server, and then added to an existing community. The object has the capability to establish a community and take on the role of administrator. During the initial phase, artificial transactions are created for a specific period, to gather basic information about the community's members and calculate trust scores. All the members of the cluster send their trust metrics (trust level, capacity and sociability) to the SIoT server and the server processes and stores these values in a list. A node whose value is minimum declares itself as the administrator and broadcasts this message to its peers and the SIoT server. On-off attacks are prevented by this model.

   b) *Distributed*: The authors of [40] have suggested a SIoT trust platform. The components of trust are knowledge, recommendations and reputation. Knowl-

edge is trust-related data offered by a trustee to assess its trust score. Knowledge is made up of four components: experience, community of interest, honesty and cooperation. Recommendations are viewed as the trustee's assessment of the trust from the perspective of trustor-related entities. An entity's reputation is created by the opinions of other entities in the system. The reputation system comprises of three modules-reputation measurement and evaluation, propagation and maintenance. Knowledge is measured using Fuzzy logic based algorithm and reputation systems are used to determine recomemmendation component and reputation component of trust. Over all trust is measured through a Utility theory based algorithm.

c) *Semi-Distributed*:The authors in [46] outlined an innovative architecture for automatically determining and upgrading a entitie's trustworthiness without the assistance of a trustworthy third party. In the decentralised environment of the SIoT, homomorphic encryption is used to preserve the node's privacy. After a transaction is complete, the communicating nodes send their feedback to Central Bulletin Board (CBB). The CBB is accessible to everyone, and anyone may read its data and determine an object's trust score. To ensure that each participant follows the protocol honestly, each device's trust score is regularly updated depending on its past trust value and recent count of votes from its peers in the network. The authors created their own Java prototype for evaluation using the cryptography toolkit BouncyCastle.

4) *Trust Update:* Trust value is updated either using event driven or time driven approach.

a) *Event Driven*: A TMS based on bipartite graph was presented in [29]. By predicting the most dependable SP for every SR, this approach lowers the risk of trustworthy nodes coming into contact with malicious nodes. Hellinger distance is employed to form a network by connecting SRs and SPs. A node's trustworthiness is determined by using node's centrality and similarity metrics. The current trust score is updated using an event-driven approach. Lastly, matrix factorization technique is used to figure out reliable nodes and eliminate cold start issues. An actual SIoT application scenario is utilised to evaluate how effectively the suggested trust management solution works and suggested technique is resilient to various network threats.

b) *Time Driven*: The authors of [34] introduced a TMS that uses ML algorithms to find malicious nodes. In this model trust features are dynamically selected based on the attack context, ANN algorithm is used for trust aggregation and time driven approach is used to update the trustworthiness score. This method gives more weight to the most recent trust values because older trust values for a device degrades over time. A node's trustworthiness is updated by direct observations, past trust efficacy, and recommendations. This model was evaluated on a real-world network (laptops and mobiles) and the data from online social networks like Twitter, Facebook and Quora are used. It can detect BSA, BMA, SPA, OSA, and DA attacks.

B. Context-aware TMS

Context refers to information that can describe (for instance, when, where, and how) an entity's condition. Trust models that consider context information during determination of trust score of a node are called Context-aware TMS. This section reviews numerous context-based trust management models, as indicated in Table III.

The authors in [63] considered context awareness, social relations and QoS constraints while assessing a node's trustworthiness. Both centralised and user-level trust management are employed. Based on criteria specified by the SR, the QoS manager determines the services and SR's feedback is stored. The context manager records the service's context being provided by SP, models it, generates reasoning, and finally publishes it. A rule-based collaborative filtering strategy is employed to determine an appropriate SP by evaluating SP's trust parameters. This model is assessed using Google and Amazon data sets on a specially created environment. The authors in [62] presented a context-aware TMS for service delegation for SIoT. In this model, trustworthiness is evaluated by considering competence, willingness and social relationships in SIoT. This model is robust and resilient to trust based attacks.

The authors in [57] have suggested a computational model of trust that focuses on extracting specific trust attributes. A node's trust score is comprised of direct observations and indirect trust. Direct trust is measured by applying ML techniques to retrieve and aggregate the following features: friendship similarity, community of interest, incentive, and node's cooperativeness and the result is stored in the database. Indirect trust is measured by collecting the reputation of a node where the trustor node and trustee node share atleast one common friend. Final trustworthiness score is determined by combining direct trust metric and indirect trust metric. To assess the model, sigcomm-2009 data set were used and K-means clustering algorithms were used to

TABLE III. Context based Trust Management Models for SIoT

Ref#	Trust Features	Aggregation	Update	Propagation	Context	Performance Metrics	Tools
[56]	Community interest, Friendship, Gain, Damage and Cost	Weighted Sum	Event driven	Distributed	Task type and Environment (Amicable and Hostile)	Success rate, Unavailable rate	Texas Instruments Z-Stack, node devices with CC2530
[57]	Friendship similarity, Reputation, Community of Interest, Cooperativeness, Reccomendation	ML based algorithm	Time driven	Distributed	Task type	Accuracy	Simulation
[58]	Friendship Similarity, Community and relations, Expected QoS, Contextual feedback, Advertised QoS	Weighted Sum	Event driven	Distributed	Time, Task type, Location	Success rate, Resiliency, Mean Absolute Error	C#, SWIM mobility model
[59]	Community of Interest, Similarity of interest	Weighted Sum	Event driven	Distributed	Time	Convergence, Transaction success rate	OMNET++
[60]	Friendship, Community of Interest, Object profile, Credibility	Weighted Sum	Time driven	Centralised	Time	Accuracy	Weka ML tool
[27]	Cooperativeness, Recomendation, Credibility, Community of Interest	Fuzzy Logic	Time driven	Distributed	Time, Location	Acuuracy, Satisfaction rate	Netlogo simulator
[61]	Social similarity, Credibility, Reputation	Weighted Sum	Event and Time driven	Distributed	Priority of service, Providers'Residual energy, Time of query	Convergence, Successful Transaction Rate	Matlab
[30]	Social relationships	Weighted Sum	Event and Time driven	Distributed	Location	No. of trusted links	SWIM, Matlab, Gephi tool
[62]	Competence, Willingness, social relationship	Weighted Sum	Event driven	Distributed	Task type	Reliability, Success rate of service	Netlogo Simulator, MATLAB
[63]	Community of Interest, Friendship, Social Contact	Weighted Sum	Event driven	Local and Centralised	Time, Location, Type of service, Activity, Capability	Reliability, Availability, Latency	Specifically designed environment

classify the nodes as benevolent nodes, malevolent nodes and neutral categories.

The authors in [56] considered mutual trust between trustor and trustee, both the parties perform the prior and the post-evaluation for one another. Past experiences and the Context are used for pre-evaluation of both the parties and the task is given to the most potential SP. Post-evaluation is performed after delegation action using the results and the environment. After outsourcing the services, a post-evaluation is carried out based on the outcomes and the environment. Real-world social networks like Facebook, Twitter, and Google+ are used to evaluate this approach.

The authors of [27] suggested a trust model that can be used to build trust between SIoT devices on their own. A node's trust score is comprised of two components- similarity trust (centrality, community of interest) and familiarity trust (historical interaction, recommendations). Utilizing direct and recommendation of a node, the familiarity trust is computed. Both internal and external similarities are used to calculate similarity trust. The small world in motion mobility model was developed by the authors with 100 users and simulations performed using the NetLogo simulator.

The work in [59], presented trust management model based on objects' common interests and object similarity. As per the trustor's preferred interests, the trustee is accessible under the proposed system. A node's trustworthiness score is comprised of direct trust metric and indirect interest-based experience. Each type of interest's scores is taken into consideration to determine a trustee's global trust, which is then used to determine direct trust score. The trustor asks potential recommenders for suggestions to evaluate indirect trust. A five-interest SIoT architecture and an OMNET++ simulator were used to assess the suggested technique.

C. Blockchain-based TMS

This section, various trust models based on blockchain-based trust management are reviewed as presented in Table IV.

The authors in [64] employed distributed ledger based consortium blockchain technology to store and retrive trust related data like the interaction histories and dynamic relationships between objects. Because some IoT objects lack the computing and storage capacity for blockchain synchronisation and administration, this blockchain technology is employed. The three different categories of nodes in the system are SR, SP and Agent. The service request, such as gathering traffic data or performing sensing activities, will be broadcast by the SR. The service will be offered to the SR by the SP. There are three processes in a trust management scheme: Creating interaction histories, recording interaction outcomes on the blockchain by agents, and determining trustworthiness. The SP's trustworthiness is determined using the social relationship between the SP and SR, the service assessment score, the number of transactions and the timing of transactions and it is updated immediately by the TMS. The suggested methodology enhances the accuracy and security of trust administration in SIoT as well as it predicts and validates the behavior of objects.

The authors in [65] have suggested a simple blockchain-based trust management mechanism for resource constrained objects for SIoT environment. The devices used in IoT are mostly resource constrained with respect to computational power and energy consumption therefore a lightweight algorithm with less calculation process was designed. IoT devices communicate with blockchain to assess their trustworthiness, and the blockchain then stores the results. This system enables multiple owners, allowing any device to be listed under more than one owner. The owner list, friends list, and counsellor list are the three lists that each device contains. Direct and indirect trust are employed to compute the trustworthiness of a node. Depending on the social connections among Iot nodes and their prior experiences, indirect trust is determined for each device using a counsellor list. This proposed framework is implemented using a Ethereum based private blockchain and it performs well in terms of cost effectiveness and the accuracy of malicious node detection.

To prevent tampering or compromise of the feedbacks utilised in trust evaluation, in [67] a blockchain-based TMS was proposed. This system also solves the problem of resource-constrained IoT nodes during the storing and computation of trust computations. A node with comparatively greater computing and storage capacity is referred to as a mobile edge node. These nodes are in charge of determining whether IoT nodes within their range are trustworthy. Smart contracts are employed in BBTM system to calculate and verify trust. This system performs well with respect to convergence, trust accuracy, and resistance to attacks like ballot stuffing and badmouthing.

The authors in [68] have suggested a blockchain-based approach for managing trust that would efficiently calculate trust values, store them safely, and allow for sharing inside the blockchain. Trustworthiness of each node is computed as a combination of direct and indirect trust where direct trust is computed depending on node's competence, cooperativeness and community of interest and indirect trust is computed based on credibility and recommendations. Multichain blockchain technology allows registered users to access the blocks. It uses Round Robin (RR) consensus method to approve transaction. This framework achieves

TABLE IV. Blockchain based Trust Management Models for SIoT

Ref#	Trust Features	Aggregation	Update	Propagation	Methodology	Blockchain type	Performance Measures	Tools
[64]	Interaction Score, Relationship factor	Weighted Sum	Event driven	Distributed (BC)	Distributed ledger technology	Consortium	Reliability, Security, Convergence time	Simulation
[65]	Direct trust, Devices and Owner friendship similarity, Ownership similarity, Social tie of node, Feedback of node	Weighted Sum	Event driven	Distributed (Ethereum BC)	Smart contract(Trust Evaluation Algorithm), Proof of Concept	Private	Success Rate(detecting untrustable nodes), Accuracy and Cost-efficiency	Ganache (private Ethereum BC), ego-Twitter dataset
[66]	Direct trustReputation, Cooperativeness, Community interest	Weighted Sum	Event driven	Distributed (Ethereum BC)	Proof of Concept, Smart contracts	Private	Privacy, Resiliency, Availability, Transparency	Simulation
[67]	Direct trust, Feedback, CPU performance, Storage capacity and Energy status, Context	Weighted Sum	Event driven	Distributed (Bitcoin BC)	Hash-based Proof of Work (POW) Distributed Consensus protocol	Permissioned	Trust accuracy, Convergence, Resiliency	Private Bitcoin BC
[68]	Cooperativeness, Competence (Energy, Comp. ability), Community of Interest, Credibility, Recommendations	Weighted Sum	Event driven	Distributed (BC)	Consensus algorithm	Private	Transparency, Integrity, Authenticity, Authorization	Multichain, NS3 simulator
[69]	Service trust, Service monitoring, Service rating, Peer task participation, Peer integrity checking	Dynamic weighing	Event and Time driven	Distributed (BC)	Lightweight and Trust based Consensus protocol	Public	Trustworthiness, Reliability, Resiliency	Simulation
[70]	Direct trust, Recommendations	Weighted Sum	Time driven	Distributed (BC)	Consensus algorithm(proof of concept), Smart contracts	Consortium	Latency, Thoughput	Hyperledger Composer, Caliper

high performance in enhancing security features (transparency, integrity, authenticity, authorization, tamper proof), reliability and resiliency to attacks.

The authors in [69] developed an improved trust model which works on a multi-layer adaptive and trust-based weighting mechanism. Each end user assesses the trustworthiness of their peers, and the findings are communicated to the blockchain as blockchain transactions, where they are tamper-proofedly recorded. To select a peer to serve as a block maker, the trust consensus mechanism is employed. The block maker selects transactions from its peers having average or high trust scores, sorts these transactions depending on trust metric categories, and applies weighted average to aggregate trust before creating a new block. A trust score request is raised by a service consumer, which triggers the smart contracts and initiates trust calculation and these current trust scores are retured to the requestor. This system also adopts incentivization process to encourage SPs' active involvement across the network. By using blockchain and smart contracts the proposed system is adaptable and reliable. Additionally, it is resistant to attacks such as ballot stuffing and slandering (badmouthing).

For supply chain management, the authors in [70] introduced Trustchain, a blockchain-based architecture for managing trust. It addresses the significant issues with integrity and traceability in the supply chain. In this approach, interactions between supply chain parties are tracked using consortium blockchain. Trustworthiness of a node is calculated by considering reputation scores and application specific features. Supply chain entitie's reputation score are determined by smart contracts by using supply chain event transactions stored on the blockchain. These calculations are secure, efficient, transparent and automated. This system achieves low latency and throughput.

D. Edge based Trust Management Models for SIoT

Due of the frequent interactions among the social objects, the SIoT scenario generates a lot of data flow. Edge computing for SIoT shifts the expense of processing and storing the data from the SO to edge servers that are nearby and have adequate storage and computational capacity, thereby reducing transmission delay and power consumption. There are studies that examine edge computing in IoT in the existing literature [71]. However, edge computing in the SIoT in its infancy, this section reviews the few articles that have been written about trust management in edge-based SIoT.

A clustering reputation model for edge based SIoT environment was presented in [72]. When a SO looks for a resource and discovers a reliable partner who possesses it, the two SOs can collaborate to finish a transaction, and once the transaction is complete, each of them gives opinion about the provider to the edge. A node's trust score is comprised of feedback from trustor and trustee, the resource's economic value, the frequency of interactions and trustor reputation. The edge node calculates and updates the new trust values for both SOs and issues updated group membership certificate. Additionally, the ledger controlled by the cloud agent is updated with these certificates. The edge agent associated with each edge domain executes a K-means clustering algorithm to categorise SOs into groups (good and malicious) and this model was verified using a simulation environment.

A lightweight trust model was presented in [73] for edge nodes in Industrial IoT. The central authority to assess device's trustworthiness joining the network rests with edge nodes. It determines the trust score of a device using compatibility, cooperativeness, delivery ratio, and recommendations. Additionally, a time-based certificate is given to the assessed node depending on the determined trust values, which may have either long or short time duration. Each node's trust value is saved in the trust agent's database. This approach was evaluated using Contiki Cooja simulator and the results demonstrate higher delivery ratio, decrease of latency with time and resilient to bad and good mouthing, whitewashing and self-promoting attacks.

The authors in [74] presented a trust model called *EdgeTrust* for nodes which are unable to carry out complex calculations. The components of this model are distributed edge devices and centralized edge clouds. Edge nodes send a trust calculation request to the closest central authority when they want to calculate the trust value of a certain node. Initially friendliness of that node is calculated, if it is high the reliability and cooperativeness is computed by the edge cloud and sent to edge node. The edge node aggregates the received parameters with locally stored previous experience to form the final trust score. Nodes with trust score greater than 0.7 are considered as trustworthy. If a node's friendliness factor is low then trust is calculated by collecting recommendations from edge cloud, thus obtained trust score should be greater than 0.9 for a node to be considered trustworthy. Simulation tool was used to evaluated this model. It is resilient to on-off attack, SPA, BSA, BMA and minimised energy consumption.

IV. Trust Based Attacks in SIoT

Malicious or misbehaving devices attack SIoT systems in order to distrupt SIoT network operations. The following are trust-based attacks executed by hostile devices that are broadly classified as coordinated and solitary attacks.

1) *Coordinated Attacks*: A set of entities work together to launch an attack. These entities have the power to either

improve a malevolent node's reputation or damage a trustworthy node's reputation.

a) *Ballot Stuffing Attack (BSA)*: Many objects cooperate to enhance the reputation of malevolent and dishonest entity by constantly giving positive feedback, thus enhancing the likelihood of picking the malevolent entity as a SP [10].

b) *Bad Mouthing Attack (BMA) or Slandering*: By spreading bad feedback on a well-behaved node, several malevolent nodes collaborate to damage its reputation. This results in reduced opportunities for a good node to be selected as a service-supplier [6].

2) *Solitary Attack*: These attacks are launched by an individual entity in the network. Initially, these nodes try to increase their reputation and on being chosen as a SP they render poor services.

a) *Self Promoting Attack (SPA)*: An object continually offers positive feedback for itself and highlights its importance in an effort to enhance its reputation and get selected as a SP. When chosen based on reputation, the node can potentially offer unsatisfactory service [6].

b) *On/Off (O/F) or Traitors attack*: A hostile entity establishes a high good reputation by behaving appropriately at first, so becoming one of the trusted entities, and then begins malicious activity. When this malicious entity's reputation falls below a certain level, it begins to perform honestly and accurately. This cycle repeats [10].

c) *White Washing Attack (WWA)*: An object removes itself from the application and then reappears, washing away its poor reputation [6].

d) *Discriminatory Attack (DA)*: An individual object provides high grade services to a set of objects while providing lesser quality services to other groups. This discrimination might take one of two forms: positive or negative discrimination. When there is negative discrimination, all nodes are given high-quality services, with the exception of the nodes who are intentionally served with poor service. In the form of positive discrimination, a individual node provides excellent service to a small number of carefully chosen entities while offering poor service to the others [10].

e) *Opportunistic Service Attack (OSA)*: An object notices that its reputation is deteriorating because of its poor service quality, it launches an opportunistic service attack to improve its reputation. When its reputation rises, it takes advantage of the opportunity to work with other malevolent objects and participates in BSA or BMA [6].

V. Strengths, Weakness and Applications of TMS

The strengths, weakness and application of process-bassed, context-based, blockchain-based and edge-based TMS are discussed in this section.

Process-based TMS

1) The strengths of process-based TMS are as follows:

a) *Improved Security*: TMS can contribute to improving SIoT system security by evaluating the trustworthiness of network nodes and other connected objects. This can help to prevent malicious attacks and lessen the chance of data breaches.

b) *Dynamic Adaptation*: TMS can adapt to changing environments and dynamic network conditions. This can make the system more responsive to changes and improve its overall performance [34].

c) *Distributed*: TMS can enable collaboration among nodes and devices in the SIoT network, this aids in enhancing the system's efficiency and lowering the chances of single point failures [40], [44].

2) The weakness of process-based TMS are as follows::

a) *Contextual Information not Considered*: These systems does not consider the contextual information like the task type, environment (hostile, amicable), location, time etc. which describe an entity's condition or an environment where an object interacts with. When contexual information is unconsidered as trust component it could compromise the security of the SIoT system.

3) The applications of process-based TMS are as follows:

a) *Social Networking*: TMS can be used to secure social networking applications, such as Facebook, Twitter, and LinkedIn, by evaluating the trustworthiness of social entities in the network. This can help to prevent malicious attacks, such as phishing and social engineering, and improve overall system security [75].

b) *Smart Communities*: Smart communities that leverage SIoT technologies employ TMS to enhance the standard of living of their residents. For instance, a smart community could use trust management sys-

tems to evaluate the trustworthiness of its members and enable secure sharing of resources [76].

Context-based TMS

1) The strengths of context-based TMS are as follows:

   a) *Enhanced Security*: A context-based TMS can help enhance the security of the SIoT by identifying trustworthy devices and users. By using contextual information, such as location, time, and activity, the system can make more precise decisions about whether to trust a device or user, reducing the risk of unauthorized access or malicious attacks. [63].

   b) *Accuracy*: These systems can accurately determine the trustworthiness of nodes based on node's behavior and contextual factors. This may reduce the possibility of security breaches and avoid unauthorised access [57], [60].

2) The weakness of context-based TMS are given below:

   a) *Context inconsistencies*: The accuracy and consistency of contextual information can be challenging to achieve due to the dynamic nature of the SIoT environment. The context information might change rapidly or become inconsistent with the actual situation, resulting in erroneous trust assessments.

   b) *Communication overhead*: Context-based trust management systems require exchanging contextual information between nodes, which can result in high communication overhead. This raises the nodes' energy use and decreases the SIoT system's overall efficacy.

3) The applications of context-based TMS are as follows:

   a) *E-commerce*: Trust in e-commerce transactions may be managed using context-based TMS. These systems can assess the user's trustworthiness and lower the risk of fraud by taking into account numerous contextual elements that include the user's purchase history and payment behaviour [58].

   b) *Healthcare*: Context-based trust management systems can be used to manage access control in healthcare environments. These systems can take different contextual elements into consideration, which include the patient's identification and the kind of medical data being viewed, to guarantee that only authorised individuals obtain access to sensitive medical information [77].

   c) The other applications are real time traffic monitoring [56], service discovery and composition [55], real world application [57].

Blockchain-based TMS

1) The strengths of blockchain-based TMS are as follows:

   a) *Decentralization*: These systems are decentralized, meaning that no one entity has authority over the system. This can make the system more secure and less vulnerable to attacks or corruption [78].

   b) *Transparency*: Transactions on the blockchain are transparent and immutable, which can help to build trust between parties. It can also increase accountability and lower the possibility of fraud [66], [68].

   c) *Security*: These systems ensures the integrity of the data stored on the blockchain utilising cryptography and consensus methods. This may increase the system's security compared to centralized systems [64].

   d) *Resilient to trust based attacks*: These systems are resilient to trust based attack types like BSA and BMA [67]

2) The weakness of blockchain-based TMS are as follows:

   a) *Scalability*: Blockchain-based trust management systems can be slow and expensive to operate, particularly as the blockchain's users and transaction volume grows.

   b) *Complexity*: Blockchain technology can be complex, and many users might not completely comprehend how it functions. This can make it difficult to adopt blockchain-based trust management systems on a large scale.

   c) *Risk of error*: While the blockchain is immutable, errors can still occur if incorrect data is entered into the system. This can lead to trust-related issues if the incorrect data is used to make decisions.

   d) *Resource Constrained Devices*: For low power and low compute devices, these systems are not appropriate.

3) The applications of blockchain-based TMS are as follows:

   a) Supply chain management: Blockchain-based TMS are employed to track and verify the authenticity of products and materials as they move through the supply chain. This can aid in lowering the risk of counterfeit goods and increase transparency and trust

between parties [78], [70].

b) Healthcare: Access control and secure medical data exchange between healthcare professionals and patients may be managed via blockchain-based TMS. This may contribute to better patient outcomes and lower the danger of data breaches [79], [80].

Edge-based TMS

1) The strengths of edge-based TMS are as follows:

a) *Improved Security*: The resource constrained IoT devices cannot execute computationally intensive security algorithms, instead they are processed and stored locally at the closest edge node of the network, which can reduce the likelihood of data breaches and unauthorised access [74].

b) *Low latency*: Edge-based TMS can reduce the latency of trust-related functions, as the trust evaluation can be performed locally on the edge nodes rather than being transmitted to a central authority [73].

c) *Low Energy Consumption*: As most of the computations are processed on the edge node, the IoT devices are not drained up [74].

2) The weakness of edge-based TMS are as follows:

a) Scalability: Edge-based TMS may not be able to scale to support large numbers of users or devices, as the trust evaluation process may become too resource-intensive.

3) The application of edge-based TMS are as follows:

a) *Health-care*: The edge-based TMS are utilised in air-quality monitoring and analysis system in the smart city environment, as well as the personal health monitoring and management system [81].

b) *Autonomous vehicles*: Edge-based TMS are employed to secure autonomous vehicle systems by evaluating trust locally on the edge nodes. This can contribute to enhancing the dependability and safety of autonomous vehicles and minimize the chances of accidents. Example secure parking allotment in smart cities [51].

VI. Issues in SIoT

The design and implementation of Trust Management must address a number of difficult problems brought on by SIoT characteristics. These challenges must be taken into account for the TMS implementation to be successful.

1) *Accuracy:* The TMS create a more secure and promising platform by classifying the nodes as trusted and malicious nodes. The trust mechanisms employed must produce more accurate classification and ban the dishonest nodes and establish trust between interacting nodes.

2) *Scalability*: The exponential rise of SIoT has resulted in the interconnection of billions of devices, which results in increased transmission of data on the network. To be truly functioning, trust models must scale along with the increasing number of device [3].

3) *Adaptability:* It allows devices to adjust to changes in their environment as well as user demands.

a) *Device heterogeneity*: With the new generation of technology and communication standards, SIoT connects all types of devices. The processor speed, storage space, battery life, operating system, protocol architecture, and other characteristics of the devices may vary. When adopting a trust model, these factors must be taken into consideration.

b) *Network heterogeneity*: SIoT services and applications overlap numerous domains. SIoT offers a standardised method to connect various networks, benefiting consumers with smart service consumption. Trust models should deal with the ideas of intra-domain and inter-domain.

c) *Dynamicity*: In the SIoT, devices and services might be accessible or inaccessible at any time. In order to continue providing their services, devices might also quit or join other network. TMS needs to consider the SIoT network's dynamic nature [3].

4) *Node Capabilities (NC):* Some of the node capability that may be considered in designing effective and efficient TMS systems are:

a) *Device energy*: The majority of the SIoT devices run on batteries. The quantity of information they process directly relates to their energy. Therefore, efficient schemes must be developed to move much of the data processing to the higher end (servers, nodes with high processing capabilities) thereby preserving the device energy.

b) *Computational and Storage Capacities (CSC)*: The cost of computation and communication in SIoT networks is the primary problem. The processing and storage capacity of SIoT devices may be constrained. Low power device's performance is directly impacted by high computing costs. Additionally, as communi-

cation costs rise, the network's efficiency in using its bandwidth and other resources, including channels, suffers. Some mechanism need to be devised to cut these expenses.

5) *Resiliency*: TMS includes managing and computing trustworthiness of a node based on direct observation and indirect recommendations. This procedure can be hampered by insider attackers that behave intentionally for their own gain or to degrade system performance. TMS should be resilient to such trust-based attacks.

VII. Research Challenges and Future Directions

This section proposes the challenges and potential research directions of the SIoT environment. The goal is to offer research directions to aid researchers in conducting additional study and improving trust models, protocols, and frameworks. The challenges related to trust management process, context-based TM, blockchain based TM, network characteristics and attacks are discussed below:

A. Challenges of SIoT Environment

1) *Challenges in Trust Management Process:* The difficulties encountered throughout each stage of the trust management process are:

a) *Choosing of Trust Features or Trust Composition*: Devices in the IoT must be able to trust one another for secure transmiting of data. The selection of the appropriate trust characteristics determines the accuracy and effectiveness of TMS. Trustworthiness of a node is aggregation of several trust parameters (QoS and social trust). Therefore, choosing the right parameters without compromising privacy is a challenge.

b) *Definition of Trustworthiness factor or Trust Formation*: Table II summarises various trust management system. The problem is to identify the exact proportion of direct trust metrics and indirect trust metrics that must be considered, so that a more accurate trust score is obtained for calculating a node's trustworthiness score.

c) *Trust Aggregation*: Most of the earlier attempts have used a weighted sum method for aggregation of trust values, despite the fact that this strategy has significant disadvantages. During trust aggregation each trust parameter is assigned a weight, which signifies the importance of each parameter. Existing mechanisms are unable to identify the trust feature that has the greatest influence on the trust score because trust features are given different weights in each scenario. Therefore, new mechanisms have to be designed for more accurate trust aggregation.

d) *Trust Update*: The trust management process uses event driven method and time driven method to update trust scores. If trust updates take long interval to update then we might be working on the old trust value or if we update very frequently then it increases the processing over head. It is important to decide on more approprite time interval to update the trust values.

e) *Initial Trust Values*: Some mechanisms has to be developed to solve the issue of newly joined user's initial trust values being inaccurately assigned because of very few interactions or no transaction history [44].

2) *Challenges in Context Based Models:*

a) *Privacy, Security and Trust*: Context-aware computing has struggled with this issue from the outset. Context has the benefit of providing more insightful information that will aid in our understanding of a scenario or set of data. Additionally, it heightens security risks because potential context misappropriation such as identity, location of a service etc.

b) *Context Discovery*: There must be a way to automatically comprehend the sensor data generated by the sensors and the relevant context after we link the sensors to the software solution. Trust models should be equipped to collect various forms of data from the information produced by smart social IoT devices.

c) *Acquisition, Modelling, Reasoning and Distribution*: It is clear that no single technique would meet the needs of the SIoT after analysis of acquisition, modelling, and reasoning from various angles. Multiple strategies have been used and integrated with encouraging results in the field. As a result, it can be challenging to anticipate when and where to use each strategy.

d) *Context Sharing*: Since different middleware solutions created by different parties will be used to connect to sensors, collect, model, and reason context, device interoperability, frameworks, and systems will become more challenging. Sharing context information between various types of middleware solutions or different occurrences with same middleware solution is crucial.

3) *Challenges in Blockchain Based Trust Management:*

a) *Storage*: As more social objects are added to the network, the frequency of transactions also increases. Each time a fresh transaction is handled, each node

contributes information to the ledger. The blockchain must keep this increasing transacting history with precision, to maintain high trust levels. Therefore, the system experiences storage issues.

b) *Scalability*: All transactions on the blockchain network should go through a validation process, due to the enormous number of transactions waiting in the queue, transactions often have to wait a lengthy period for validation. As network size increases the response time also increases. Hence scalability-related concerns require further investigation.

c) *Context Awareness*: A device may function well for a resource-light service while oscillating for other heavy services. In a multi-service SIoT environment, it's critical to pay attention to context-awareness when computing trust.

d) *Trust Incentivization*: To emphasize on the power of end users and to overcome monopoly of some nodes in service provision, efficient trust incentivization mechanisms have to be developed. These mechanisms should encourage peers to take part in various community activities in an active and trustworthy manner. They should also contain penalties for inactive nodes or improper behaviour.

4) *Challenges Based on Network Properties:*

a) *Capability of Devices*: All SIoT applications cannot use previous trust management techniques since SIoT devices have different degrees of computing resources, storage space, communications protocols, operating systems, and I/O channels. The trust management algorithms should take into consideration the capabilities of these devices.

b) *Dynamic Nature of Network*: The evolving nature of SIoT network acquires new devices while the older ones are removed. Furthermore, a device's dynamic character, including its membership, patterns of interaction, network architecture and location changes must be considered by TMS algorithms.

c) *Heterogeneity*: IoT networks are made up of various kinds of device with varying levels of processing power, storage capacity, and energy. When contrasted to a limited power device, a device with great computing power may easily trick the trust computation. Additionally, the process of trust calculation is impacted by the heterogeneity of networks because devices from other networks may be less valid than devices from the same network. When developing their trust models, very few researchers have taken heterogeneity into account. Any application where several different devices connect over diverse networks must take heterogeneity into account.

5) *Attack Related Challenges:* In literature, majority of the trust managemet systems have considered to clasify benovelent and malicious nodes but little focus is spent on building a system resilient to trust based attacks. More intelligence has to be embedded to find out misbehaving nodes and trace the pattern for misbehaving nodes.

B. Future Research Directions for TMS in SIoT

This section covers the unresolved research problems that must be resolved for the effective use of TMS for SIoT systems. This study identifies areas for further research and points researchers in the right path for creating a trust model that meets their needs. Table V provides an outline of the TMS's examination of the issues it addresses. The proportion of efforts put forward by the studied TMS in relation to various issues is depicted in Figure 3. According to the survey, most schemes have placed a greater emphasis on resilience against BMA and BSA while placing less attention on node scalability and residual energy.

1) Choosing of Trust Features or Trust Composition

Trustworthiness of a node should be composed as dynamic component which varies its composition according to its environment (services or applications specific).

2) Trust Aggregation

As weighted sum aggregation methods has several drawbacks, researchers have proposed using ML-based aggregation to determine the weights of each trust parameter in terms of its significance [57]. However, the computing cost of these techniques increases the computational delay. Therefore, ML algorithms that are computationally efficient has to be developed.

3) Trust Propagation

Security algorithms that consider the propagation of trust values between nodes, especially if the SP and SR belong to different clusters has to be developed.

4) Trust Update Mechanisms

For effective deployment of TMS, it is crucial to choose the right update window size. Whenever there is a longer updating window period, then nodes will calculate trustworthiness of a node with older values. In contrast if the updating window period is very short it may increase computation and processing costs. Hence, security solutions should be developed to address this problem.

TABLE V. Assessment of trust models based on issues in TMS

TMS Type	Ref#	Accu.	Scal.	Adapt.	NC		Resiliency						
					Energy	CSC	SPA	BMA	BSA	OOA	OSA	DA	WA
Process-based TMS	[38]	✔	✗	✔	✗	✗	✔	✔	✔	✔	✗	✗	✗
	[39]	✗	✔	✔	✗	✗	✗	✔	✔	✔	✔	✗	✗
	[42]	✗	✗	✗	✗	✗	✗	✗	✗	✔	✗	✔	✗
	[41]	✔	✔	✗	✗	✔	✗	✗	✗	✔	✗	✗	✗
	[29]	✔	✔	✗	✗	✔	✗	✗	✗	✗	✔	✗	✔
	[45]	✗	✗	✗	✗	✔	✔	✔	✔	✔	✔	✔	✔
	[47]	✗	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✗
Context-based TMS	[56]	✗	✗	✔	✗	✗	✔	✗	✗	✗	✔	✗	✗
	[57]	✔	✗	✗	✗	✗	✗	✔	✔	✗	✗	✗	✗
	[58]	✔	✗	✗	✗	✗	✔	✔	✔	✔	✗	✗	✗
	[59]	✔	✗	✗	✗	✗	✔	✔	✗	✗	✗	✔	✔
	[27]	✗	✗	✗	✗	✗	✗	✔	✗	✗	✗	✗	✗
	[61]	✗	✗	✗	✔	✗	✔	✔	✔	✗	✔	✗	✗
	[62]	✗	✗	✔	✗	✔	✔	✔	✔	✗	✔	✔	✔
Blockchain-based TMS	[64]	✗	✗	✔	✗	✗	✔	✔	✔	✗	✗	✗	✔
	[65]	✔	✗	✗	✗	✔	✗	✔	✔	✗	✗	✗	✗
	[66]	✗	✗	✗	✗	✗	✗	✔	✔	✗	✗	✗	✗
	[67]	✗	✗	✗	✗	✔	✗	✔	✔	✗	✗	✗	✗
	[68]	✗	✗	✗	✔	✔	✗	✔	✔	✔	✗	✗	✗
	[69]	✗	✗	✔	✗	✗	✗	✔	✔	✗	✗	✗	✗
	[70]	✗	✗	✔	✗	✗	✗	✔	✔	✗	✗	✗	✔

Accu.: Accuracy; Scal.: Scalability; Adapt.: Adaptability

5) Initial Trust Values

The TMS cannot categorize the newly joined nodes as benign or malicious as they have very few interaction. In most of the trust models new nodes are initialized with a neutral value like 0.5. This makes the system vulnerable to white washing attack. Very little research is done to eliminate this problem [44].

6) Mobility

As the IoT devices are movable in nature, which results in considerably more compute power and energy. Hence we require a strong resource management mechanism to solve it.

7) Context Awareness

In the literature, a number of context-aware TMSs have been presented as shown in Table III suggesting different contexts which are generally time, location and task type.

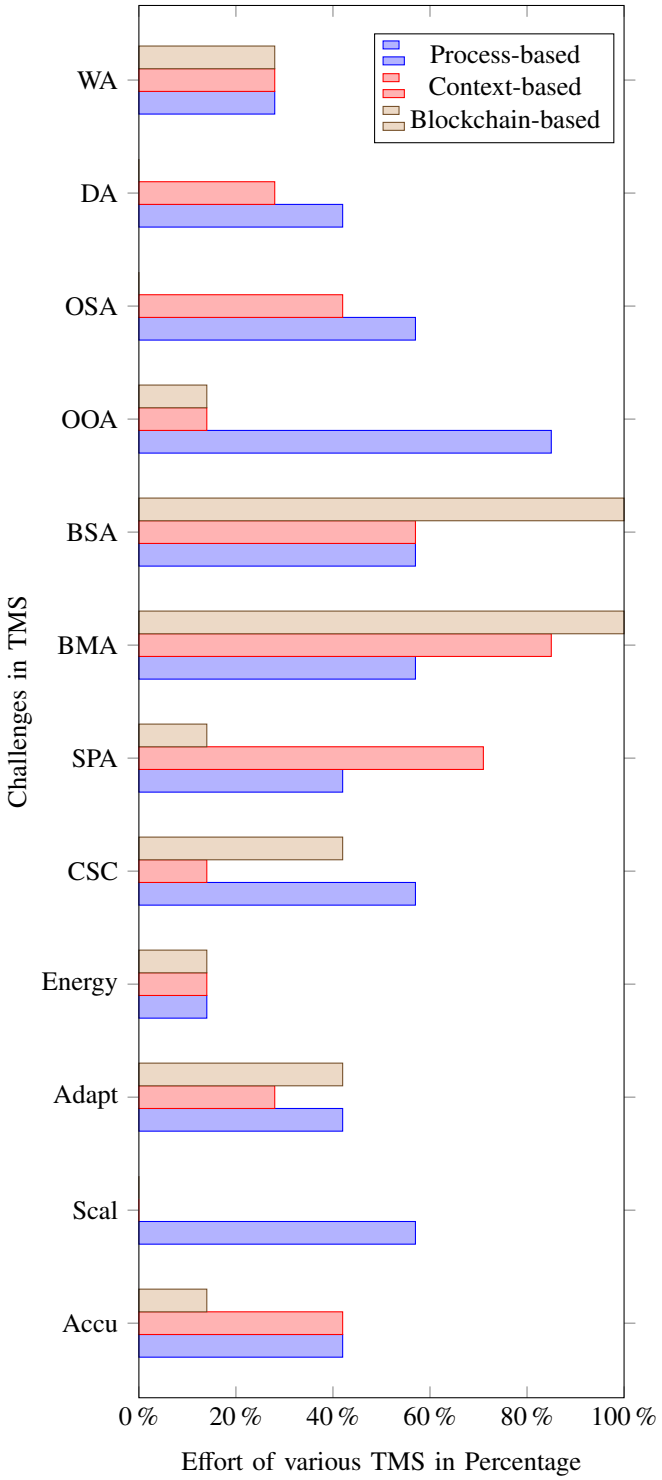

Figure 3. Proportion of efforts made by various TMS categories to address SIoT issues.

Other context such as residual energy, environmental condition and temporal information need to be considered.

8) Capabilities of Devices

There is a need for simple and lightweight trust management protocols since it might be difficult for restricted IoT devices with limited processing capabilities to calculate trust score.

9) Resilience towards Trust Related Attacks

The resiliency of TMS is not taken into account in most works. Consequently, it is necessary to develop security algorithms that can recognise the malicious activity patterns of adversary nodes.

VIII. Conclusions

TMS plays an essential role to guarantee secured and efficient deployment of IoT applications and services by taking into account the uncertainties involved with device interaction. This survey has reviewed the existing TMS by classifying them as process-based TMS, context-based TMS, blockchain-based TMS and edge-based TMS and also their strengths, limitations and applications are discussed. Analyses of current TMS schemes in comparison to their trust features, aggregation methods, trust update, trust propagation approach, evaluation tool, and performance metrics is presented. Further, researchers are presented with research challenges and open future directions to create an appropriate trust model in accordance with their application needs.

References

[1] N. Srinidhi, E. Nagarjun, J. Shreyas, S. Dilip Kumar, and D. Chouhan, "Ensuring fault tolerant connectivity in iot networks," in *Computer Communication, Networking and IoT*. Springer, 2021, pp. 391–400.

[2] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 5, pp. 1253–1266, 2013.

[3] M. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Social internet of things (siot): Foundations, thrust areas, systematic review and future directions," *Computer Communications*, vol. 139, pp. 32–57, 2019.

[4] M. Rashmi and C. V. Raj, "A review on trust models of social internet of things," *Emerging Research in Electronics, Computer Science and Technology*, pp. 203–209, 2019.

[5] R. K. Chahal, N. Kumar, and S. Batra, "Trust management in social internet of things: A taxonomy, open issues, and challenges," *Computer Communications*, vol. 150, pp. 13–46, 2020.

[6] W. Z. Khan, S. Hakak, M. K. Khan *et al.*, "Trust management in social internet of things: Architectures, recent advancements, and future challenges," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7768–7788, 2020.

[7] M. Malekshahi Rad, A. M. Rahmani, A. Sahafi, and N. Nasih Qader, "Social internet of things: vision, challenges, and trends," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–40, 2020.

[8] S. Sagar, A. Mahmood, Q. Z. Sheng, J. K. Pabani, and W. E. Zhang, "Understanding the trustworthiness management in the social internet of things: A survey," *arXiv preprint arXiv:2202.03624*, 2022.

[9] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9326–9337, 2019.

[10] A. I. A. Ahmed, S. H. Ab Hamid, A. Gani, M. K. Khan *et al.*, "Trust and reputation for internet of things: Fundamentals, taxonomy, and open research challenges," *Journal of Network and Computer Applications*, vol. 145, p. 102409, 2019.

[11] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, "Towards trustworthy internet of things: A survey on trust management applications and schemes," *Computer Communications*, 2020.

[12] L. Kumar and N. Badal, "An effective method of hybrid encryption on iot."

[13] L. Atzori, A. Iera, and G. Morabito, "Siot: Giving a social structure to the internet of things," *IEEE communications letters*, vol. 15, no. 11, pp. 1193–1195, 2011.

[14] V. Miori and D. Russo, "Improving life quality for the elderly through the social internet of things (siot)," in *2017 Global Internet of Things Summit (GIoTS)*. IEEE, 2017, pp. 1–6.

[15] K. H. Rahouma, R. H. Aly, H. F. Hamed *et al.*, "Challenges and solutions of using the social internet of things in healthcare and medical solutions—a survey," in *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*. Springer, 2020, pp. 13–30.

[16] B. Jadhav and S. Patil, "Wireless home monitoring using social internet of things (siot)," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. IEEE, 2016, pp. 925–929.

[17] P. Kumaran and R. Sridhar, "Social internet of things (siot): Techniques, applications and challenges," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. IEEE, 2020, pp. 445–450.

[18] N. Gulati and P. D. Kaur, "Towards socially enabled internet of industrial things: architecture, semantic model and relationship management," *Ad Hoc Networks*, vol. 91, p. 101869, 2019.

[19] L. A. Maglaras, A. H. Al-Bayatti, Y. He, I. Wagner, and H. Janicke, "Social internet of vehicles for smart cities," *Journal of Sensor and Actuator Networks*, vol. 5, no. 1, p. 3, 2016.

[20] M. Roopa, A. Siddiq, R. Buyya, K. Venugopal, S. Iyengar, and L. Patnaik, "Dynamic management of traffic signals through social iot," *Procedia Computer Science*, vol. 171, pp. 1908–1916, 2020.

[21] G. Delnevo, R. Girau, C. Ceccarini, and C. Prandi, "A deep learning and social iot approach for plants disease prediction toward a sustainable agriculture," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7243–7250, 2021.

[22] C. K. Panda and R. Bhatnagar, "Social internet of things in agriculture: an overview and future scope," *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*, pp. 317–334, 2020.

[23] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, pp. 1–33, 2013.

[24] J. Kraciuk and T. Li, "Trust and the internet of things," in *Trust, Digital Business and Technology*. Routledge, 2023, pp. 187–201.

[25] A. Meena Kowshalya and M. Valarmathi, "Dynamic trust management for secure communications in social internet of things (siot)," *Sādhanā*, vol. 43, no. 9, pp. 1–8, 2018.

[26] S. E. A. Rafey, A. Abdel-Hamid, and M. Abou El-Nasr, "Cbstm-iot: Context-based social trust model for the internet of things," in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*. IEEE, 2016, pp. 1–8.

[27] H. Xia, F. Xiao, S.-s. Zhang, C.-q. Hu, and X.-z. Cheng, "Trustworthiness inference framework in the social internet of things: A context-aware approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 838–846.

[28] H. Xiao, N. Sidhu, and B. Christianson, "Guarantor and reputation based trust model for social internet of things," in *2015 International wireless communications and mobile computing conference (IWCMC)*. IEEE, 2015, pp. 600–605.

[29] S. Aalibagi, H. Mahyar, A. Movaghar, and H. E. Stanley, "A matrix factorization model for hellinger-based trust management in social internet of things," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[30] U. S. Premarathne, "Mag-siot: A multiplicative attributes graph model based trust computation method for social internet of things," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2017, pp. 1–6.

[31] J. Guo, R. Chen, and J. J. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.

[32] S. Mon, S. G. Winster, and R. Ramesh, "Trust model for iot using cluster analysis: A centralized approach," *Wireless Personal Communications*, pp. 1–22, 2021.

[33] A. Khanfor, A. Hamrouni, H. Ghazzai, Y. Yang, and Y. Massoud, "A trustworthy recruitment process for spatial mobile crowdsourcing in large-scale social iot," in *2020 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 2020, pp. 1–6.

[34] S. Babar, P. Mahalle *et al.*, "Trust management approach for detection of malicious devices in siot," *Tehnički glasnik*, vol. 15, no. 1, pp. 43–50, 2021.

[35] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013.

[36] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in internet of things: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2017.

[37] M. Knappmeyer, S. L. Kiani, E. S. Reetz, N. Baker, and R. Tonjes, "Survey of context provisioning middleware," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1492–1519, 2013.

[38] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," *IEEE transactions on dependable and secure computing*, vol. 13, no. 6, pp. 684–696, 2015.

[39] E. Kokoris-Kogias, O. Voutyras, and T. Varvarigou, "Trm-siot: A scalable hybrid trust & reputation model for the social internet of things," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Ieee, 2016, pp. 1–9.

[40] N. B. Truong, T.-W. Um, and G. M. Lee, "A reputation and knowledge

based trust service platform for trustworthy social internet of things," *Innovations in clouds, internet and networks (ICIN), Paris, France*, pp. 104–111, 2016.

[41] O. B. Abderrahim, M. H. Elhdhili, and L. Saidane, "Tmcoi-siot: A trust management system based on communities of interest for the social internet of things," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 747–752.

[42] A. M. Kowshalya and M. Valarmathi, "Trust management for reliable decision making among social objects in the social internet of things," *IET Networks*, vol. 6, no. 4, pp. 75–80, 2017.

[43] C. A. Kerrache, N. Lagraa, R. Hussain, S. H. Ahmed, A. Benslimane, C. T. Calafate, J.-C. Cano, and A. M. Vegni, "Tacashi: Trust-aware communication architecture for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5870–5877, 2018.

[44] S. Ding, Z. Yue, S. Yang, F. Niu, and Y. Zhang, "A novel trust model based overlapping community detection algorithm for social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2101–2114, 2019.

[45] C. Marche and M. Nitti, "Trust-related attacks and their detection: A trust management model for the social iot," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3297–3308, 2020.

[46] M. A. Azad, S. Bag, F. Hao, and A. Shalaginov, "Decentralized self-enforcing trust management system for social internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2690–2703, 2020.

[47] W. Abdelghani, I. Amous, C. A. Zayani, F. Sèdes, and G. Roman-Jimenez, "Dynamic and scalable multi-level trust management model for social internet of things," *The Journal of Supercomputing*, vol. 78, no. 6, pp. 8137–8193, 2022.

[48] C. Goswami, R. Raman, B. G. Pillai, R. Singh, B. Dhanne, and D. Kapila, "Implementation of a machine learning-based trust management system in social internet of things," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2022, pp. 1586–1590.

[49] K. A. Awan, I. U. Din, A. Almogren, and J. J. Rodrigues, "Autotrust: A privacy-enhanced trust-based intrusion detection approach for internet of smart things," *Future Generation Computer Systems*, vol. 137, pp. 288–301, 2022.

[50] S. Sagar, A. Mahmood, K. Wang, Q. Z. Sheng, J. K. Pabani, and W. E. Zhang, "Trust–siot: Towards trustworthy object classification in the social internet of things," *IEEE Transactions on Network and Service Management*, 2023.

[51] J. Ali and M. F. Khan, "A trust-based secure parking allocation for iot-enabled sustainable smart cities," *Sustainability*, vol. 15, no. 8, p. 6916, 2023.

[52] J. Byabazaire, G. O'Hare, and D. Delaney, "Data quality and trust: Review of challenges and opportunities for data sharing in iot," *Electronics*, vol. 9, no. 12, p. 2083, 2020.

[53] W. Najib, S. Sulistyo *et al.*, "Survey on trust calculation methods in internet of things," *Procedia Computer Science*, vol. 161, pp. 1300–1307, 2019.

[54] N. B. Truong, T.-W. Um, B. Zhou, and G. M. Lee, "From personal experience to global reputation for trust evaluation in the social internet of things," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[55] Z. Chen, R. Ling, C.-M. Huang, and X. Zhu, "A scheme of access service recommendation for the social internet of things," *International Journal of Communication Systems*, vol. 29, no. 4, pp. 694–706, 2016.

[56] Z. Lin and L. Dong, "Clarifying trust in social internet of things," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 234–248, 2017.

[57] S. Sagar, A. Mahmood, Q. Z. Sheng, and W. E. Zhang, "Trust computational heuristic for social internet of things: A machine learning-based approach," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[58] M. Khani, Y. Wang, M. A. Orgun, and F. Zhu, "Context-aware trustworthy service evaluation in social internet of things," in *International Conference on Service-Oriented Computing*. Springer, 2018, pp. 129–145.

[59] S. Talbi and A. Bouabdallah, "Interest-based trust management scheme for social internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1129–1140, 2020.

[60] O. B. Abderrahim, M. H. Elhedhili, and L. Saidane, "Ctms-siot: A context-based trust management system for the social internet of things," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1903–1908.

[61] B. Jafarian, N. Yazdani, and M. S. Haghighi, "Discrimination-aware trust management for social internet of things," *Computer Networks*, vol. 178, p. 107254, 2020.

[62] L. Wei, J. Wu, C. Long, and B. Li, "On designing context-aware trust model and service delegation for social internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4775–4787, 2020.

[63] S. Javaid, H. Afzal, F. Arif, N. Iltaf, H. Abbas, and W. Iqbal, "Catswots: Context aware trustworthy social web of things system," *Sensors*, vol. 19, no. 14, p. 3076, 2019.

[64] L. Wei, J. Wu, and C. Long, "Enhancing trust management via blockchain in social internet of things," in *2020 Chinese Automation Congress (CAC)*. IEEE, 2020, pp. 159–164.

[65] M. Amiri-Zarandi, R. A. Dara, and E. Fraser, "Lbtm: A lightweight blockchain-based trust management system for social internet of things," *The Journal of Supercomputing*, pp. 1–19, 2022.

[66] M. Amiri-Zarandi and R. A. Dara, "Blockchain-based trust management in social internet of things," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2020, pp. 49–54.

[67] X. Wu and J. Liang, "A blockchain-based trust management method for internet of things," *Pervasive and Mobile Computing*, vol. 72, p. 101330, 2021.

[68] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for iot," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–8.

[69] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain and trust for secure, end-user-based and decentralized iot service provision," *IEEE Access*, vol. 8, pp. 119 961–119 979, 2020.

[70] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trustchain: Trust management in blockchain and iot supported supply chains,"

in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 184–193.

[71] L. Fotia, F. Delicato, and G. Fortino, "Trust in edge-based internet of things architectures: state of the art and research challenges," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–34, 2023.

[72] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. Sarnè, "A social edge-based iot framework using reputation-based clustering for enhancing competitiveness," *IEEE Transactions on Computational Social Systems*, 2022.

[73] I. U. Din, A. Bano, K. A. Awan, A. Almogren, A. Altameem, and M. Guizani, "Lighttrust: lightweight trust management for edge devices in industrial internet of things," *IEEE Internet of Things Journal*, 2021.

[74] K. A. Awan, I. Ud Din, A. Almogren, H. A. Khattak, and J. J. Rodrigues, "Edgetrust: A lightweight data-centric trust management approach for iot-based healthcare 4.0," *Electronics*, vol. 12, no. 1, p. 140, 2022.

[75] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[76] J. E. Kim, X. Fan, and D. Mosse, "Empowering end users for social internet of things," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 2017, pp. 71–82.

[77] M. Ebrahimi, M. S. Haghighi, A. Jolfaei, N. Shamaeian, and M. H. Tadayon, "A secure and decentralized trust management scheme for smart health systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1961–1968, 2021.

[78] G. D. Putra, C. Kang, S. S. Kanhere, and J. W.-K. Hong, "Detrm: Decentralised trust and reputation management for blockchain-based supply chains," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2022, pp. 1–5.

[79] E. M. Abou-Nassar, A. M. Iliyasu, P. M. El-Kafrawy, O.-Y. Song, A. K. Bashir, and A. A. Abd El-Latif, "Ditrust chain: towards blockchain-based trust models for sustainable healthcare iot systems," *IEEE access*, vol. 8, pp. 111 223–111 238, 2020.

[80] N. K. Al-Shammari, T. Syed, and M. Syed, "An edge–iot framework and prototype based on blockchain for smart healthcare applications," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7326–7331, 2021.

[81] J. Yuan and X. Li, "A reliable and lightweight trust computing mechanism for iot edge devices based on multi-source feedback information fusion," *Ieee Access*, vol. 6, pp. 23 626–23 638, 2018.

Santhosh Kumari received the B. E and M. Tech degrees in Computer Science and Engineering in 2008 and 2013 from Visvesvaraya Technological University Belgaum, India. She worked as a faculty member in the department of Computer Science and Engineering at Dr. T. Thimmaiah Institute of Technology from 2010 to 2020 for 8 years. She is currently pursuing PhD degree in Computer Science and Engineering at University of Visvesvaraya College of Engineering (UVCE), Bangalore. Her current research interest include Internet of Things(IoT), Social Internet of Things(SIoT) and Trust Management.

Dr. S. M Dilip Kumar received the B. E, M. Tech, and Ph. D degrees in Computer Science and Engineering in 1996, 2001, and 2010 respectively. He is currently the Professor in the Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. He has guided eight Ph. D candidates in Computer Science and Engineering. He has published 110 papers in peer reviewed International Journals and Conferences. He was the Principal Investigator for a research project in the area of grid computing sponsored by the Science and Engineering Research Board, Department of Science and Technology (SERB-DST), Government of India. Another research project sponsored by SERB-DST in the area of the Internet of Things is ongoing. He has completed two consultancy projects in mobile governance and e-FMS sponsored by the Government of Karnataka. His current research is on cloud computing, fog computing and Internet of Things.

Dr. Venugopal K R is the Former Vice Chancellor, Bangalore University, Bangalore and Principal, UVCE for 15years. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D in Economics from Bangalore University and Ph.D in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 72 books on Computer Science and Engineering. He has guided 30 Ph.D students, published more than 1000 research papers, and credited with 37 Patents. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining. He is a Fellow of IEEE and ACM Distinguished Educator.