A Survey of Trust Management for Internet of Things

Konsta, Alyzia Maria; Lafuente, Alberto Lluch; Dragoni, Nicola

Published in:
IEEE Access

Link to article, DOI:
10.1109/ACCESS.2023.3327335

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 September 2023, accepted 18 October 2023, date of publication 25 October 2023, date of current version 6 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327335

SURVEY

A Survey of Trust Management for Internet of Things

ALYZIA MARIA KONSTA[1], ALBERTO LLUCH LAFUENTE[2], AND NICOLA DRAGONI[3]
DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Corresponding author: Alyzia Maria Konsta (akon@dtu.dk)

This work was supported by the Innovation Fund Denmark and the Digital Research Centre Denmark, through the bridge project "Secure Internet of Things (SIOT)—Risk Analysis in Design and Operation."

ABSTRACT Internet of Things (IoT) is a network of devices that communicate with each other through the internet and provide intelligence to industry and people. These devices are running in potentially hostile environments, so the need for security is critical. Trust management aims to ensure the reliability of the network by assigning a trust value to every node, indicating its trust level. In this paper, we systematically review and analyze the current state of the art in trust management appraoches for IoT. We provide a classification into nine categories based on the tools, methods, and technologies used to form trust management techniques (collect information for trust formation, compute, and store the trust values). We also discuss the limitations and strengths of each category, as well as the open challenges and future research directions. We aim to help the reader understand the current challenges of the field, design a solid trust management system, and navigate through the literature.

INDEX TERMS Attacks, Internet of Things, IoT, security, survey, trust evaluation, trust management.

I. INTRODUCTION

Internet of things (IoT) is a recent technology broadly used in our everyday lives. According to Cisco's annual report (2018–2023), the number of devices connected to IP networks will be more than three times the global population by 2023 [71]. The term IoT was first introduced in 1999 by Kevin Ashton in the context of supply chain management [36], but in the last decade, the concept has been used in multiple fields like agriculture [13], health care [49], energy [14] and transportation [15] among others. IoT forms a network of devices -such as RFID, sensors, mobile phones, etc.- that communicate through the internet. These devices gather information from their environment and provide intelligence to industry and people.

IoT objects are running at remote locations in potentially hostile environments, so they are vulnerable to security attacks. However, these resource-constrained devices cannot support the customary security algorithms, which require powerful hardware and software. Taking into account the magnitude of IoT and the domains using this technology, we can imagine that a huge amount of sensitive information is processed by IoT devices. Therefore, the need for security is crucial. One method used to assess the reliability of the network is trust management. Trust management aims to ensure the reliability of the network by assigning a trust value to every node, indicating its trust level. Thus, the information provided by a node with a high trust level is considered reliable. To create a trust relationship, at least two entities must be involved: the trustor and the trustee.

In this work, we present an exhaustive survey of trust management for IoT. This study provides a classification based on the methods and technologies used for trust formation. In every paper, we examine nine different dimensions, including limitations and strengths.

A. SCOPE OF THE PAPER

The scope of the paper focuses on specific aspects of trust management. In particular, how information necessary for trust is gathered, how trust is updated, how trust is formed, how trust information is propagated among nodes, which is the threat model considered, whether the approach is validated with experiments and how those are conducted, and whether tools like simulators are available to support evaluation of approaches to trust management. Further information on these aspects is provided in Section VI.

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

We decided to include these aspects of trust after examining which of them are included in the current literature, to which we added the experiment and simulator aspects since we believe it is important to consider tool-based support to assess the proposed approaches. In our perspective, it is important to examine *which* experiments were conducted and *how* they were conducted in every work. We also discuss the limitations and strengths of each paper.

B. GOAL OF THE PAPER

The goal of this survey is to identify existing trust management mechanisms and to classify them based on their main technologies in order to spot limitations in each category. We highlight limitations and research gaps by examining some key aspects of trust. Our objective is to provide an overview of the field and discuss the challenges identified. We hope that this will help interested readers navigate the vast literature in this field. We also aim to help the reader design a solid trust management system and navigate through the literature based on the method used and some key properties.

C. CONTRIBUTIONS

The main contributions of this paper are:

1) A literature review of the existing trust management techniques.
2) A categorization of the existing literature based on the techniques/tools used to form the trust management method.
3) Pointing out current challenges and future research directions for trust management in the IoT.

D. STRUCTURE OF THE PAPER

The rest of the paper is structured as follows:

- Section II: Provides the necessary background by summarizing the main concepts of IoT and trust management used throughout the paper.
- Section III: Presents the research method we have followed to structure our research and the main questions we aim to address through this survey.
- Section IV: Discusses the related works, i.e. other surveys that have considered the topic of trust management in IoT.
- Section V: Presents an overview of the categories by providing the reader with a quantification study.
- Section VI: Provides the classification of the literature and description for all the papers participating in this research.
- Section VII: Discusses the main challenges and future research directions.
- Section VIII: Summarizes and concludes the study.

II. BACKGROUND

In this section, we provide an overview of the basic concept we are going to discuss in the following chapters. The structure of this section is as follows:

- Section II-A provides an introduction to the IoT.
- Section II-B provides an overview of the main concepts of trust management.
- Section II-C recalls the classical three-layer architecture of the IoT.
- Section II-D describes the main categories used to structure our survey.
- Section II-E provides an overview of the main classes of attacks on IoT as known from the literature.
- Section II-F, Section II-G and Section II-H respectively cover three technologies that are relevant for many of the papers included in the survey, namely cloud computing, edge computing, and blockchain.

A. INTRODUCTION TO IoT

Internet of Things (IoT) refers to a network of devices that can communicate with each other, and collect and exchange data over the internet. In effect, IoT enables everyday objects to become ''smart'' enhancing their functionality and enabling new services. IoT surrounds a diverse ecosystem of devices ranging from everyday items like thermostats, and refrigerators to complex machinery, equipped with sensors for data collection.

This recent technology has a significant impact on our everyday lives and industries. The data collected by IoT devices can be analyzed to derive insights, make predictions, and optimize processes. Machine learning and AI can play a role in extracting valuable information from IoT data. IoT has a wide range of applications across various industries, including smart homes, healthcare, agriculture, manufacturing, transportation, energy management, and more. Examples include smart cities, wearable fitness trackers, and autonomous vehicles.

IoT has the potential to revolutionize many aspects of our lives and industries by providing real-time data, automation, and insights that can lead to improved efficiency, convenience, safety, and sustainability. However, it also presents challenges, such as data security. Since the number of connected devices grows drastically and IoT devices are operating in potentially hostile environments the need for security is crucial. Moreover, IoT devices are usually resource-constrained devices that cannot support the broadly used security algorithms that consume a vast amount of energy and resources. Thus, the need for alternative security mechanisms that support the characteristics of these devices arises. One mechanism that is broadly used is trust management [71], [78].

B. TRUST MANAGEMENT

In a dynamic environment such as an IoT network, it is really important to detect malicious nodes. In our everyday lives, we want to collaborate and connect with trustworthy people, both in our personal and professional lives. In the same way, IoT devices, to function smoothly, need to interact with trustworthy nodes, that provide them with honest information.

The basis of trust is the accurate identification of IoT devices. Each device should have a unique identity so that it can be distinguished in the network. With this unique identity, the device would be able to access the network and communicate with other devices. After the authentication is in place, access control mechanisms define which actions and data each device is authorized to access.

It is also crucial to ensure data integrity and confidentiality, both in transit and at rest, from unauthorized access and tampering. Trustworthy devices are essential for the smooth functionality and security of IoT networks. This includes assessing the behavior of the devices, monitoring suspicious anomalies, and revoking trust if necessary. The process starts with collecting information for other nodes, using this information to compute the trust level, and then storing and using this piece of information. The trust level determines if a node can trust or interact with another. As people, we would never close a deal with a person that we did not trust; the same applies to the IoT nodes in a network. Different trust models can be used in IoT, such as hierarchical trust models, reputation-based trust models, and trust models, depending on the specific IoT application and architecture. Trust management in IoT is an ongoing process that requires collaboration between manufacturers, service providers, and end users to establish and maintain trust throughout the life of IoT devices and services.

Trust management is associated with multiple challenges, such as the cold-start problem, which is the initialization of the trust values during bootstrapping or a new node entering a network. Sometimes privacy issues may arise since trust management mechanisms may handle sensitive data and the resources that have to be consumed for them to operate [6], [70], [71]. In Section VII we are analyzing and summarizing all the challenges identified throughout this survey.

C. IoT ARCHITECTURE

According to most researchers, IoT is a three-layer architecture [40], [48]. These are the Perception layer, the Network layer, and the Application layer.

- *Perception layer:* It consists of physical devices, such as sensors, that gather information from the environment.
- *Network layer:* This layer is responsible for connecting the devices to servers and network devices. Also, the protocols of this layer are used to transmit and exchange information among the devices.
- *Application layer:* Serves as an intermediate layer between the network and the IoT services. The data collected from the smart devices is transferred to the application layer. Applications like smart health, smart home, etc. belong to this layer.

D. CATEGORIES

One of the contributions of this work is to point out the tools, methods, and technologies being used to form trust management techniques. The categories defined represent the technologies used in each paper. We divide the papers into nine different categories: Blockchain, Context, Social, Game Theory, Probabilistic, Prediction, Fuzzy, Direct, and Recommendations. These categories represent the technologies used in every paper to gather the information for the trust formation, compute the final trust, and store the trust-related data as stated in Table 1. The rows in Table 1 represent the categories defined, and the columns the aspects of trust management to every category contributes.

TABLE 1. Categories used in this work.

Category	Info Gathering	Computation	Storing
Blockchain		✓	✓
Context		✓	
Social		✓	
Game Theory		✓	
Probabilistic		✓	
Prediction		✓	
Fuzzy		✓	
Direct	✓		
Recommendations	✓		

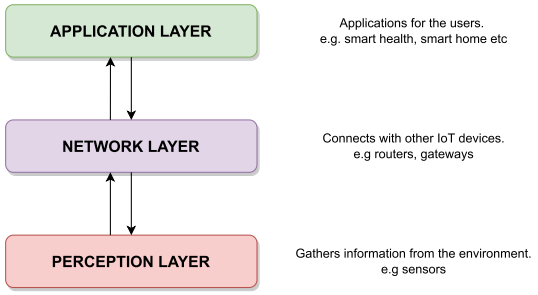

FIGURE 1. IoT architecture.

To form a trust management technique one can combine the tools offered by any category; thus, each paper could in principle be part of more than one category.

We are going to examine seven different dimensions in every category to identify the research gaps: information gathering, experiments, trust propagation, threat model, trust update, trust formation, and the simulator. We summarize some limitations and strengths for each dimension in Table 2. Following we introduce the seven dimensions of trust:

1) INFORMATION GATHERING

One trust management system should collect data to execute the trust computation. Information Gathering is the first step toward trust computation. It refers to the process of collecting knowledge about the trust parameters. We can divide information gathering into two categories:

- *Direct Trust:* Trust is formed based on direct observations and interactions between the two parties involved in the trust relationship.
- *Indirect Trust:* The trustor and the trustee do not share any previous interactions. Trust is formed based on the recommendations of other nodes [38], [70].

2) EXPERIMENTS
We are also going to examine what kind of experiments took place in every work.

3) TRUST PROPAGATION
Trust propagation refers to how the trust evidence is propagated to the nodes involved in the system, and it is divided into two categories:
- *Distributed:* Every node stores the trust values of the other nodes. The nodes interact and exchange trust evidence with other nodes. The nodes independently store and compute the trust values without involving a central authority in the procedure.
- *Centralized:* A central authority is present to compute and store the trust values. This entity is responsible for propagating and handling the trust evidence [38], [70].

4) THREAT MODEL
It is a set of attacks examined in each work. The malicious node can perform these attacks in the system under investigation.

5) TRUST UPDATE
Trust update refers to when the trust is updated and can be divided into two categories:
- *Event driven:* When a specific event triggers the trust update. For example, when a node requests an interaction, trust update can be triggered.
- *Time driven:* When the trust is being updated in time intervals [38], [70].

6) TRUST FORMATION
Trust formation refers to how the overall trust is formed. It is divided into two categories:
- *Single Trust:* Only one trust parameter is considered to form the trust of a node.
- *Multi Trust:* Several parameters are used to form the trust of a node since the trust is considered multidimensional [38], [70].

7) SIMULATOR
Refers to the simulator used in every work to perform the experiments. Of course, in some cases, real devices were used.

E. ATTACKS
Based on the examined literature, an IoT node can perform the attacks presented in Table 3. An attack can be trust-related or belong to a different layer of the IoT architecture.

In an IoT system using a trust management technique where the nodes are evaluated, the trust level of the node plays an important role in their image. A node with a good trust level can have multiple collaborators and influence in the system. Hence, we are concerned with trust-related attacks. First, we are going to elaborate on these kinds of attacks, identified so far in the literature [38], [67]:
- *Bad mouthing attacks (BMA):* A malicious node can provide bad recommendations for an honest node, trying to ruin its reputation. The goal of this attack is to lower the reputation of an honest node.
- *Ballot stuffing attacks (BSA):* A malicious node is providing good recommendations for other malicious nodes. The goal of this attack is to increase the trust level of other malicious nodes, thus increasing their influence in the network.
- *Self-promoting attacks (SPA):* A malicious node can provide good recommendations for itself to increase its influence in the network.
- *Opportunistic service attacks (OSA):* A malicious node can provide good service to gain a high trust level and then cooperate with other malicious nodes to perform bad-mouthing and ballot-stuffing attacks.
- *On-Off attacks (OOA):* A malicious node can provide sometimes bad and sometimes good services. With this attack, a node avoids being labeled as an untrustworthy node.

In the scope of this paper, we are also going to discuss the following types of attacks mentioned in the literature included in this survey:
- *Eclipse attack (EA):* In this type of attack, the malicious node isolates the victim from the rest of the network [27].
- *Node Capture attack (NCA):* This type of attack targets the physical devices of the IoT, in terms of communication links, fake data input, etc. [48]
- *Replay attack (RA):* In this type of attack, the malicious node is listening to a communication to gain information and misdirect the receiver. [48] For example, if Alice shares a piece of information with Bob (to prove her identity), then Eve is eavesdropping on the conversation and stores the information Alice shared. Now Eve can maliciously communicate with Bob and pretend to be Alice.
- *Sybil attack (SA):* A malicious node has multiple identities and can place itself simultaneously in different places in the network.
- *Whitewashing attack (WA):* When a node with a bad reputation is re-entering the network with a different identity to reset its reputation.
- *Denial of Service attack (DoS):* When a malicious node is sending multiple requests to the network to make it unavailable for the rest of the users.
- *Spoofing attack (SFA):* When a node uses a different identity and pretends to be someone else.
- *Blackhole attack (BA):* When a node is deleting all messages, it is supposed to forward. This attack is creating a gap in the network.
- *Wormhole attack (WHA):* The nodes involved in this attack are stronger nodes that communicate at longer distances. The packets are forwarded from one malicious node to the other through a tunnel. In this way, they can

TABLE 2. Limitations and strengths for each dimension.

Dimension	Limitations	Strengths
Direct Trust	• A malicious node might act malevolent to other nodes.	• Based on your own experience.
Indirect Trust	• Some nodes might provide false recommendations. • Need for filtering recommendations.	• Provides a global view of the nodes' behaviors.
Experiments	• No limitation on adding experiments.	• Experiments can support the methodology and its functionality. • Key metrics prove how the method improves existing approaches.
Distributed	• More resources are consumed, for every node to store and compute the trust values.	• Every node holds its values and no central authority can provide false data.
Centralized	• A central authority can provide false data. • Static procedure.	• The computational and storage weight is taken away from the nodes.
Threat Model	• No limitation on stating the threat model.	• Understand where every method is applicable. • A key metric to compare approaches.
Event-Driven	• Very often updates can lead to more energy consumption.	• The trust is updated after every interaction and a malicious node will be caught immediately.
Time-Driven	• For long intervals, a malicious node can act for a certain time.	• A nice balance between frequency and saving resources can save a lot of energy.
Single-Trust	• One view of trust.	• Saving energy.
Multi-Trust	• More energy is needed to compute all parameters.	• Multidimensional view of trust.
Simulator	• No limitation on stating the setup environment.	• Details about the setup of an experiment can hold a lot of information (e.g. simulation vs. real data).

TABLE 3. Category of every attack.

Attack	Trust	Application	Network	Perception
BMA	✓			
BSA	✓			
SPA	✓			
OSA	✓			
OOA	✓			
EA				
NCA		✓		
RA				✓
SA			✓	
WA				
DoS			✓	
SFA			✓	
BA			✓	
WHA			✓	
IA		✓		
SDA				

trick the other nodes of the network into believing that these two nodes are closer.

- *Injection attacks (IA):* A malicious code is injected to disturb the smooth functioning of the network.
- *Sleep deprivation attack (SDA):* The malicious node is making frequent requests to a node to keep it awake and consume all the battery resources quickly [61].

F. CLOUD COMPUTING

Cloud computing enables users and devices to store and process huge amounts of data with the use of services provided on the internet. Some of the benefits of cloud computing:

- Cost: No need to acquire specialized hardware or IT teams to manage these infrastructures.
- Performance: Provide high-performance services to heterogeneous devices.
- Speed: High-speed services.
- Security: Provides services that enhance security.

For the reasons mentioned above, cloud computing can be paired with IoT devices. IoT devices collect a vast amount of information and send it to the cloud for storage and computation of different metrics. The cloud is on the network layer Figure 1. The data are collected by the perception layer and sent to the network layer, where the cloud services process the data and reform them to be useful for the users on the application layer [73].

G. EDGE COMPUTING

Cloud computing, discussed above, offers some major advancements, but it remains far away from the local network. This may result in delays, poor raw data, and no real-time insights. These drawbacks led the research community to come up with Edge Computing. This framework allows the IoT gateways to perform pre-processing on the raw data and send only the useful ones to the cloud. So, edge computing can be seen as a way to store and process data closer to data production, decreasing delays [41].

H. BLOCKCHAIN

The blockchain is a decentralized, distributed ledger consisting of a list of records called blocks. These blocks are linked

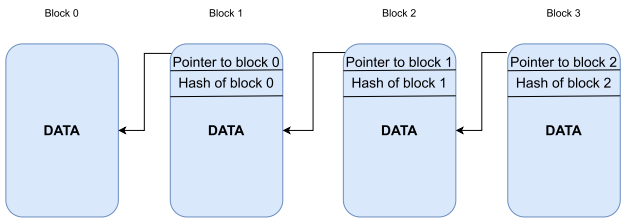

FIGURE 2. Blockchain example.

using a cryptographic hash, as seen in Figure 2. The first application of blockchain was for the Bitcoin cryptocurrency, but it has also been used in many other applications.

The most significant properties of blockchain are transparency, immutability, and decentralization.

Transparency means that all users can see other transactions only through their public addresses. Immutability means that once a piece of information is stored on the blockchain, it cannot be changed. Decentralization means that everything is distributed and there is no central authority [48].

III. RESEARCH METHOD

In this section, we present the research method we adopted to discover the existing literature on trust management in IoT. We followed the research method proposed by Petersen et al [60]. In the following, we describe the research questions, the search method, and the study selection. The structure of this section is as follows:

- Section III-A defines the research questions.
- Section III-B provides the search method we followed.
- Section III-C provides the exclusion criteria we followed during study selection.

A. RESEARCH QUESTIONS

This paper aims to study the existing literature on trust management in IoT, so we focus on the following research questions:

- **RQ1:** Which methods are currently used in the field?
- **RQ2:** What is the threat model of those proposals?
- **RQ3:** What are the strengths and limitations of each proposal?
- **RQ4:** What are the challenges and future directions?

B. SEARCH METHOD

We used the PICOC criteria [45] to come up with relevant keywords for our search:

- **Population:** We are interested in works in IoT nodes.
- **Intervention:** We are interested in works that propose a trust management technique.
- **Comparison:** We compare different kinds of trust management schemes in the IoT based on design features, security capabilities, and performance.
- **Outcomes:** We present trust management techniques for the IoT: opportunities and limitations, as well as future challenges.
- **Context:** We are interested in any paper that proposes a trust management technique for IoT nodes.

Based on the above criteria, we came up with the following keywords: "Internet of Things", "IoT", "trust", "trustworthy" and "node". We performed our search in DTU Findit https://findit.dtu.dk, which is an open (guest access) database that includes publications from widely known journals and databases: Elsevier, IEEExplore, ACM Digital Library, etc. We used the following query: title:(IoT OR "internet of things") AND title:(trust OR trustworthy) AND abstract:(node). Our search returned 341 papers. The final pool of papers was selected based on the study selection we describe in the next paragraph.

C. STUDY SELECTION

We started with 341 papers after identifying the duplicates, we applied the following exclusion criteria:

- **E1:** The full text of the paper is not available.
- **E2:** The papers in not provided in English.
- **E3:** The work is not focusing on IoT.
- **E4:** The work is not focused on trust management techniques.

After applying the above exclusion criteria, we also performed the snowballing [77] technique and ended up with 53 papers, which constitute the final pool.

IV. RELATED WORK

In this section, we present some papers conducting surveys on trust management for the IoT. We also present a table with useful information regarding the papers. On Table 4 we included 8 columns. The first column indicates the paper under examination. The rest of the columns refer to different kinds of characteristics of each paper. When a characteristic denoted by the corresponding column is satisfied, we fill out a ✓. The second column, Trust Attacks, describes if a paper takes into account trust-related attacks and the third column describes if the paper takes into account other types of attacks. The fourth column denotes if a paper is classifying the papers under study based on the underlying technology used in each one to produce the trust results. The fifth column indicates if a paper is published after 2020. The sixth column General indicates if the paper is conducting a general survey on trust management or focusing on a specific method or field. The seventh column, Future Directions denotes if a paper is proposing future directions for the research community. Finally, the last column indicates if the paper proposes some solution for the challenges identified.

Guo et al. [38] classified trust computation models for service management in IoT systems. Their classification is based on the techniques used in trust composition. They proposed five design dimensions for a trust computation model: trust composition, trust propagation, trust aggregation, trust update, and trust formation. The authors mentioned the pros and cons of each dimension's options. Finally, they also identified gaps in IoT trust computation research and suggested future research directions. This work

was published before 2020, and the authors only mention trust-related attacks in their survey.

Yan et al. [78] investigated the properties of trust, proposed objectives for IoT trust management, and provided a survey of the literature on trustworthy IoT. Furthermore, the authors discussed unsolved issues and research challenges and proposed a research model for holistic trust management in IoT. To conduct holistic IoT trust management, the trust properties that impact trust relationships were explored and classified into five categories: Trustee's objective properties, Trustee's subjective properties, Trustor's subjective properties, Trustor's objective properties, and the Context in which the trust relationship resides. This paper was published before 2020, and the authors do not mention the threat models of the papers included in their survey.

Singh and Kandpal [71] discussed the fog computing three-layer architecture and state-of-the-art models. The bottom layer of Fog Computing comprises IoT devices [71]. Through their survey on trust management in Fog Computing Singh et al., also identified trust and security challenges. This recent work only focuses on Fog Computing architectures.

Kumar and Sharma [48] focused on research regarding trust using Blockchain technology in the IoT environment. The authors pointed out some challenges and issues of trust management in IoT and proposed Blockchain-based solutions. Furthermore, some issues regarding the integration of Blockchain with IoT were discussed. Finally, a comparative analysis between traditional and Blockchain-based trust management techniques was presented. Kumar et al. only focus on blockchain-related research.

Sharma et al. [70] presented the different stages involved in the process of trust management. Furthermore, the authors presented a survey on trust management schemes. The survey is conducted considering direct observations and indirect recommendations, distributed, semi-distributed, centralized schemes, and blockchain-based schemes for trust management in IoT. Moreover, they provided a comparative study of the existing schemes based on some system parameters like the computation model, input attributes, evaluation tools, and performance metrics, examining their strengths and weaknesses. The paper also highlights open research challenges and presents future directions for the researchers. Sharma et al. do not provide classification based on the technologies used and focus only on trust-related attacks.

Alshehri and Hussain [6] were focused on scalable and context-aware trust management for the IoT. They present the concept of IoT and the importance of trust. The authors also provided a comparative evaluation of existing trust solutions for the IoT focusing on scalability. Also, they presented a trust management protocol for the IoT. Furthermore, the authors also provided a context-aware evaluation of the IoT and compared the different trust solutions. Finally, the authors gave some future directions for research. This work was published before 2020, and the authors do not provide the threat models of the works included in this survey. Also, this work focuses only on context-aware trust management.

Saeed et al. [67] proposed a classification tree in this survey for trust management models. The classification scheme takes into account five dimensions of trust. They do not examine the experiments conducted in every work or the threat model mentioned in every category. The authors examine some trust-related attacks on IoT devices. Finally, they point out some future directions. This work only mentions trust-related attacks; They do not classify the papers based on the technologies used.

Pourghebleh et al. [62] presented a survey where the selected techniques were categorized into four main classes, including recommendation-based, prediction-based, policy-based, and reputation-based. The authors also present a discussion where they compare the literature based on some metrics, such as accuracy, adaptability, availability, heterogeneity, integrity, privacy, reliability, and scalability. Furthermore, some future challenges and directions are provided. Our work has more updated literature, covering papers published since 2020, while [62] covers only papers published until 2019. As a result, our work covers a total of 36 papers that were not covered before. Moreover, while [62] focuses only on trust-related attacks, Our work considers additional classes of attacks at the application, network, and perception level. We argue that a well-designed trust management system should be capable of dealing with all kinds of attacks. Indeed, this position is shared by some of the surveyed papers, which consider threat models and concrete attack methods that are not directly trust-related.

The summary of our findings is presented in Table 4. We can observe from the Table that most of the works refer only to trust-related attacks. To identify the weaknesses of the field, one should also take into account the non-trust-related attacks that are being tackled in the literature.

Until now, only [71] refers to the technologies used to classify the papers but focuses only on trust-related attacks. Also, only [48] provides some solutions to the challenges presented, but the survey is focused on Blockchain-based trust management techniques.

We can conclude that our work provides a more complete view of general trust management techniques. We point out the technologies used in every work and provide a categorization based on these technologies. We also take into account all kinds of attacks and provide future directions and a solution to support the design of a multidimensional trust management system.

V. OVERVIEW OF THE CATEGORIES

In this section, we will provide a comprehensive picture of the categories and attacks mentioned. More specifically, we will conduct a quantification study and discussion on attacks, publishers, and publication years. The structure of this section is as follows:

- Section V-A presents discusses the percentage of papers in each category.
- Section V-B presents data related to the trust model.

TABLE 4. Comparison with related work.

Paper	Trust Attacks	Other Attacks	Methods' Classification	After 2020	General	Future Directions	Solutions
[38]	✓				✓	✓	
[78]					✓	✓	
[71]	✓		✓	✓	✓	✓	
[48]	✓	✓		✓		✓	✓
[70]	✓			✓	✓	✓	
[6]						✓	
[67]	✓			✓	✓	✓	
[62]	✓				✓	✓	
Our work	✓	✓	✓	✓	✓	✓	✓

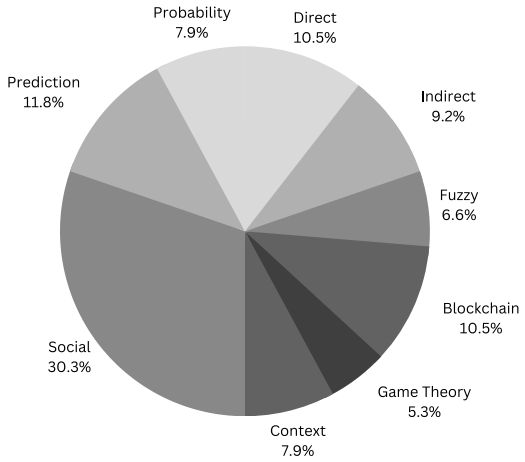

FIGURE 3. Percentage of papers classified in each category.

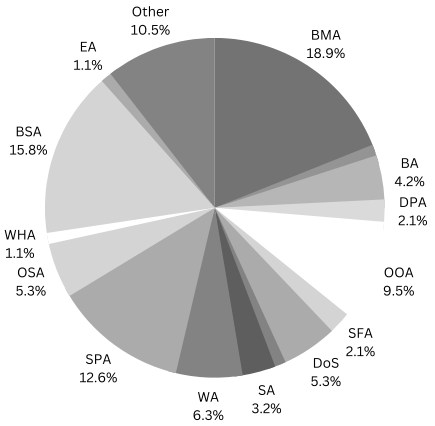

FIGURE 4. Percentage of every attack mentioned in every threat model.

- Section V-C provides results regarding the publisher and the publication year.
- Section V-D provides more details about the categories.

A. PERCENTAGE OF PAPERS IN EACH CATEGORY

In total, the survey consisted of 53 papers. Each paper may belong to different categories. We classified the papers into nine different categories. In Figure 3 you can see the percentage of papers classified in each category.

As we can see, the most popular category is Social. To calculate trust in this category, social aspects such as friendship are considered. Like in our everyday lives, we are more invested in trusting people that we have previously interacted with or have a social group in common. It simulates a network of nodes as a group of things with social interactions. The nodes can develop relationships with the other participants, and the level of trust is related to their social interactions. This scheme also provides the opportunity to filter the recommendations.

B. PERCENTAGE OF ATTACKS

In this section, we present some data related to the threat model provided. It is important to mention that 28 out of the 53 papers defined a specific threat model for their research. In Figure 4 you can see the statistical information regarding the attacks studied in the literature.

We observe that trust-related attacks -BMA, BSA, SPA, OSA, OOA- are studied the most in the literature. A trust management system needs to be able to deal with trust-related attacks, but this is only the foundation. A well-designed trust management system should be capable of dealing with all kinds of attacks. We can see that only 1.10% of the papers consider the EA, RA, SDA, and WHA. These kinds of attacks may cause severe damage to an IoT network.

C. PUBLISHER AND PUBLICATION YEAR

In this section, we present results regarding the publisher of every paper and the publication year. It would be interesting to see when the research community started to investigate further trust management solutions for the IoT.

We can observe in Figure 5 that IEEE is the leading publisher, with more than half papers included in this survey. Most of the papers related to trust management for IoT have been published by IEEE.

In Figure 6 we can see that the most popular year for trust management for IoT publications was 2019. The first paper published in Trust Management specifically for IoT was in

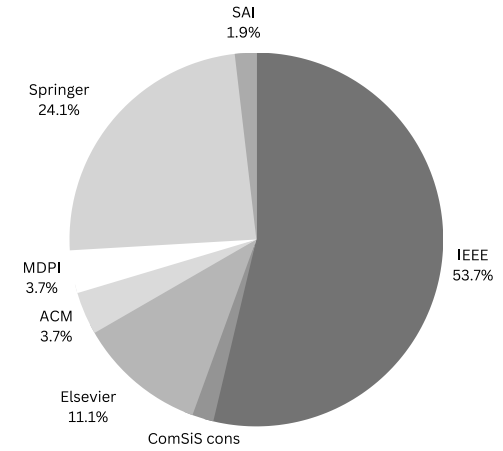

FIGURE 5. Percentage of papers published by each publisher.

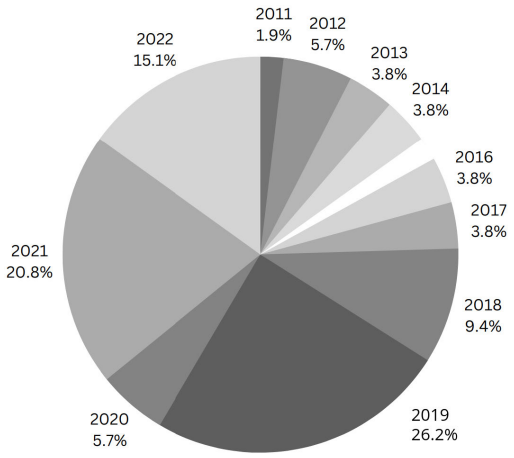

FIGURE 6. Number of papers published in each year.

2011. It makes sense since IoT is a relatively new technology that has gained a lot of attention in the last decade. The papers involved in this research were published until the end of 2022.

D. ATTACKS, TRUST PROPERTIES, EXPERIMENTS

In this section, we will give an overview of the categories identified in Table 3.

1) BLOCKCHAIN

In the Blockchain category, we included seven papers. Five out of seven papers included a threat model and trust parameters, while all of them included a section with the experiments.

The attacks that gained the most attention were the trust-related attacks BSA and BMA. Resource-constrained IoT devices are not always capable of participating in a blockchain network as foul nodes, so they are vulnerable to EA. However, EA was studied in one paper.

During the experiments, only a small percentage of the works took into account scalability, and no work took into account energy consumption. Both of these two components are important, especially for extended IoT networks.

2) CONTEXT

In this category, we included six papers. One out of six specified a threat model, five out of six defined the trust parameters, and all of them had a section with the experiments.

Context is a really important parameter for trust. Some individuals may be trustworthy only under specific circumstances and contexts. However, only one paper in this category specifies the threat model, which includes the SA and BMA. Most of the trust-related attacks are not mentioned. A solid trust management system should be able to confront trust-related attacks.

There are no experiments conducted for scalability and energy consumption. A lot of works are comparing their work with other existing models to justify the efficiency of their solution.

3) SOCIAL

In this category, we included 23 papers. Twelve out of 23 specified a threat model, 21 out of 23 defined the trust parameters, and 22 of them had a section with the experiments.

Trust-related attacks are gaining most of the attention. The experiments are focused on performance and comparison with other models. Energy consumption has taken into account a small percentage of papers, i.e. 5%.

4) FUZZY

In this category, we included five papers. One out of five specified a threat model, four out of five defined the trust parameters, and all of them had a section with the experiments.

Trust-related attacks are only mentioned in the research; other attacks are not mentioned in the threat models presented. Also, most of the works are conducting comparison experiments.

5) GAME THEORY

In this category, we included four papers. Three out of four specified a threat model, two out of five defined the trust parameters, and all of them had a section with experiments.

Many game-theoretic approaches aim to reduce the energy consumption in an IoT network. So, the trust parameters and the experiments take into account the energy factor for evaluating trust.

6) PROBABILISTIC

In this category, we included six papers. Three out of six specified a threat model, four out of six defined the trust parameters, and all of them had a section with experiments.

The papers are dealing with trust-related attacks. Two probabilistic approaches take into account Energy as a trust parameter and also in the experiments.

7) PREDICTION

In this category, we included nine papers. Five out of nine specified a threat model, while all of them defined the trust parameters and had a section with the experiments. The experiments do not include scalability and energy. In this category, some non-trust-related attacks are gaining attention, like SFA, SDA, and DoS.

VI. CLASSIFICATION BASED ON THE USED METHODS

This section provides a classification of all the papers considered in the survey. The classification uses the categories described in Section II-D. In each category, we compare the approaches proposed in the papers using several dimensions. To help the reader navigate each category we provide at the beginning of the corresponding section a table that provides an overview of the considered papers. The tables include one column per dimension: Information Gathering, Trust Update, Experiments, Centralized, Trust Formation, Threat Model, and Simulator as stated in the Background section. Additionally, the last two columns will state the Limitations and Strengths of each method. More in detail:

- Paper: The paper under examination.
- Information Gathering: Filled out with the word Direct or Both, in case the paper is using both Direct and Indirect trust. It will remain empty in case the paper does not state how the information gathering is taking place.
- Trust update: Filled out with the word Time-Driven, Even-Driven, or Both, in case the paper uses both approaches. It will remain empty if the paper does not state how the trust update is taking place.
- Experiments: If a paper provides experiments supporting the theorems a check mark (✓) will be placed, otherwise it will remain empty.
- Centralized: If a central entity is responsible for Trust Management, a check mark (✓) will be placed, otherwise it will remain empty.
- Trust Formation: Filled out with the word Single-Trust or Multi-Trust, depending on which approach the paper is using. It will remain empty if the paper does not state how the trust formation is taking place or state the trust parameters.
- Threat Model: When a paper explicitly states the threat model, a check mark (✓) will be placed; otherwise, it will remain empty.
- Simulator: When a paper states which simulator was used to conduct the experiments, a check mark (✓) will be placed; otherwise, it will remain empty.
- Limitations: Summarize the limitations of the paper.
- Strengths: Summarize the strengths of the paper.

The tables can also be used to navigate through the literature by providing the reader with key characteristics of every work. To further support the reader, Table 5, provides the overview of all the papers of the literature and their placement in the categories and dimensions.

A. DIRECT TRUST

Direct observations refer to the process of gathering information for trust calculation through direct communication between the nodes. The node relies on its own observation when the trust calculation is taking place.

A summary of the works relying only upon direct observations for trust evaluation is presented in Table 6. We can observe from the table that all the works provide experiments, prefer the multi-trust approach, and use the direct trust method - as stated by the category. We can see that 62.5% of the papers mention the simulator used. The Threat model is explicitly defined in 50% of the papers. We can also observe that 25% of the papers provide a Centralized approach. Finally, 37.5% of the approaches are Event-Driven, while 50% of the papers do not define the Trust Update procedure. In the rest of the section, we provide a summary of each paper.

Alshehri and Hussain [7] proposed a cluster-based architecture, including one super node and many main nodes that are responsible for multiple cluster nodes. Only the cluster nodes are considered malicious. The malicious nodes can perform OOA. Alshehri and Hussain introduced five algorithms to calculate the trust score of every cluster node. To calculate the trust score, they take into account the quality of service, the history score, and the trust score. They use a fuzzy approach to classify the trust scores into fuzzy sets. The nodes are then classified into 3 categories: trusted, semi-trusted, and non-trusted. Based on the trust score, the nodes can change clusters, and based on the category, they can perform specific acts. They performed experiments in the Cooja simulator regarding scalability, the accuracy of different attacks, and fuzzy and non-fuzzy, approaches and they presented diagrams with the results. The authors do not specify the trusted entities involved in the procedure. Also, they only deal with one trust-related attack. Finally, they proposed a HEXA decimal-based messaging system that can be used to detect tampered messages in transit, and they isolate the untrusted nodes from the network.

Dedeoglu et al. [28] proposed a system containing sensors and gateways. Gateways run the blockchain and are associated with several sensors. A malicious sensor can tamper with the data, and a malicious gateway can generate invalid blocks. A lightweight block generation scheme was proposed where blocks are generated at time intervals. The block validation mechanism adapts the block validation scheme based on the reputation of the node that generated the block and the number of validators. For the consensus mechanism, the following method was introduced: If a validator detects an invalid transaction, it broadcasts INVALID and the nodes have to validate the transaction; otherwise, the block is appended to the blockchain. The proposed technique evaluates the trustworthiness of sensor observations. The sensor assigns a confident value to the data and sends it to a gateway. The gateway compares the data with the data of the other cluster sensors (assumption: the

TABLE 5. Overall overview of the categories.

Property	Subvalues	Direct Trust	Recommendations	Fuzzy Logic	Blockchain	Game Theory	Context	Social	Prediction	Probabilistic
Info Gathering	Direct	[7], [28], [50], [75], [42], [44], [68], [72]		[7]	[28]		[68]	[42]	[42], [72]	[44], [73]
	Direct & Indirect		[4], [30], [64], [79], [12], [29], [55]	[23], [37], [53]	[9], [43], [46], [63]	[31], [32], [66]	[2], [3], [8], [65], [52]	[10], [54], [59], [65], [2], [25], [58], [74], [3], [16], [18], [47], [9], [11], [17], [26], [1], [19], [24], [76], [51], [52]	[5], [19], [74], [76], [1], [51], [52]	[21], [34], [35], [74]
Trust Update	Time-Driven	[44]	[4]	[33]	[33], [43], [46]	[32], [33], [66]		[10], [19], [52]	[19], [51]	[44]
	Event-Driven	[28], [50], [68]	[12], [30], [55], [64]	[23], [37]	[20], [27], [28], [63], [9]		[2], [8], [65], [68], [3], [52]	[2], [54], [59], [65], [16], [25], [47], [58], [3], [9], [17], [18], [24], [52], [76]	[5], [52], [76]	[34]
	Both			[53]		[31]		[11]		
Trust Formation	Multi-Trust	[7], [28], [50], [75], [42], [44], [68], [72]	[30], [55], [64], [79], [12], [29]	[7], [23], [37], [53]	[20], [27], [46], [63], [43]	[31]	[2], [8], [65], [68], [3], [52]	[10], [54], [59], [65], [2], [16], [58], [74], [3], [11], [26], [47], [17], [24], [42], [76], [1], [19], [51], [52]	[5], [72], [74], [76], [1], [19], [42], [51], [52]	[21], [44], [74], [75]
	Single-Trust		[4]		[9], [27]	[32]		[9]		
Experiments	-	[7], [28], [50], [75], [42], [44], [68], [72],	[4], [30], [64], [79], [12], [29], [55]	[7], [23], [37], [53], [33]	[27], [28], [46], [63], [9], [20], [33], [43]	[31]–[33], [66]	[2], [8], [65], [68], [3], [52]	[10], [54], [59], [65], [2], [25], [58], [74], [3], [16], [18], [47], [9], [17], [26], [42], [1], [19], [24], [76], [51], [52]	[5], [72], [74], [76], [1], [19], [42], [51], [52]	[21], [44], [74], [75], [34], [35]
Centralized	-	[50], [68]	[29], [30]				[68]	[10], [11]		
Threat Model	-	[7], [28], [44], [50]	[64]	[7], [33]	[27], [28], [46], [63], [9], [20], [33]	[20], [28], [43]	[65]	[16], [25], [54], [65], [9], [17], [18], [51], [1], [19], [24], [76]	[1], [5], [19], [76], [51]	[21], [34], [44]
Simulator	-	[7], [28], [44], [75], [72]	[4], [55], [64], [79], [12]	[7], [23], [37], [53]	[20], [28], [43]	[31], [66]	[2], [52]	[2], [24]–[26], [19], [51], [52]	[1], [19], [51], [72], [52]	[34], [35], [44], [75]

sensors in the same cluster have correlated data). In the end, based on the result, the reputation of the node is recalculated. The gateways store the information on a blockchain. The reputation of the gateways is calculated based on their actions during the generation and validation of the blocks. The trust parameters taken into account are the confidence of the data source, the reputation of the data source, and evidence from other observations. The experiments took place in the NS-3 simulator, and they are both blockchain-related and trust-related. The final results were presented in diagrams. One limitation of this approach is that the neighboring sensors have to gather the same category of data; otherwise, it cannot be applied.

Ma et al. [50] proposed a multi-mix attack method. The sub-attacks include: tamper, replay, and drop attacks. The nodes update their cognition when a packet is transferred. The base station collects all cognitions from nodes and performs a central trust evaluation. For detection of the malicious nodes, the node's trust should be forwarded to the k-means clustering module. The trust properties used for trust evaluation are honesty, straightness, and volume. During the experiments, the accuracy of the proposed method was tested. This paper does not deal with trust-related attacks, but it considers a type of attacker that can perform mixed attacks.

Wang et al. [75] proposed a system consisting of Mobile edge nodes (MEN) and common sensors. The MEN are connected to a small number of sensors. In this paper, a mobile edge trust evaluation scheme is proposed. The evaluation of the trustworthiness of sensor nodes is achieved using a probabilistic graph model. The probabilistic graph model is used to represent the relationship between nodes. The interaction of node i with node j can be described as P and Q. P is a positive influence of node i on node j and Q is a negative one. The information gathered for the formation of trust is the result of data collection and communication behavior. Also, a moving strategy method is proposed to decrease the travel distance MEN has to cover to evaluate every sensor. The experiments were conducted using MATLAB and NS-3 and were focused on the performance of the mechanism, the analysis of energy consumption, and the testing of the proposed moving algorithm. The results were presented in diagrams. The paper does not specify the threat model, which is a drawback. A strength, on the other hand, is that the authors propose a moving strategy for energy savings in a high-mobility environment.

Saied et al. [68] proposed a context-aware and multi-service trust management system. Upon a request from a node asking for assistance, the trust manager starts the entity selection process to return a set of trustworthy assisting nodes to the requester. A set of recommenders sends reports; the most important are those that lie to the same or more similar services and recent ones. A quality of recommendation score is assigned to each node reflecting its trustworthiness when rating other nodes. The context was used to filter out recommendations and select the most relevant ones. The trust is calculated based on the following parameters: The score is given by the requester node to the service provider evaluating the offered service, a weight that depends on time similarity, and quality of recommendations. Experiments were performed focused on the comparison of reactions against different kinds of attacks like on-off, bad-mouthing, and selective behavior attacks. The authors do not specify the threat model, which is a drawback. On the other hand, they involve the context in the trust-related procedure. A node can act differently in different contexts.

Joshi et al. [44] proposed a system that consists of several resource-constrained IoT nodes with a short radio range and a base station with a limitless source of energy as a central authority. This research work has presented a 2-state HMM with a Trusted state and a compromised state, together with essential and unessential output as observation states. The trustworthiness of the node is modeled by the 2-state HMM

to predict the likelihood of the node's next state. The state transition probability matrix is defined by the energy consumed, the number of modified packets, and the number of forwarding packets. The malicious nodes can drop the packets or tamper with the data. Experiments were conducted in MATLAB to evaluate the network's trustworthiness with various percentages of compromised nodes and compare it with other methods. The results were presented in diagrams. The authors are taking into account only two kinds of attacks. The authors are using energy consumption as a key characteristic for calculating trust. This is interesting since increased activity might be malicious, but also energy of the nodes is taken into account in a resource-constrained environment.

Subhash et al. [72] proposed the Power Trust. Power Trust assigns trust values to the nodes of the network based on energy auditing. Using the energy auditing model, they calculate the trust values of every node present in the network dynamically and predict physical and cyberattacks. To detect the attacks, a deep learning model was trained with past data that contains normal and excessive energy consumption due to an attack. The model can predict both physical and cyberattacks. The experiments were performed in the Cooja simulator, and they were focused on the performance and the accuracy of the method. The results were presented in diagrams. The authors do not state the threat model, and they do not give details about the deep learning model used. The method takes into account energy consumption, which is important in a resource-constrained environment. Finally, they also predict both physical and cyberattacks.

Jayasinghe et al. [42] proposed a system where the nodes form communities of interest. In this model, the transactions are under evaluation and should be determined if a transaction is trustworthy. The trust parameters used are co-location relationships, co-work relationships, mutuality and centrality, and cooperativeness. An unsupervised learning technique was employed to label the data's trustworthiness. After the labeling, an SVM model predicts the trust level. Experiments were performed to observe the performance of the proposed solution. The authors do not specify the threat model. They offer a metric that provides a perception of a node before interaction. This can be used when there have been no previous interactions or a new node has just entered the network.

B. RECOMMENDATIONS

To gather the information needed for the trust calculations, the nodes can ask for recommendations concerning the node under evaluation from other nodes. This procedure is also called indirect trust. There are many reasons why recommendations are valuable for trust evaluation. Some works use recommendations as a supplement to direct observations; the summary of these works is presented in Table 7. In this category, we include works that only use Recommendations as a method. Works in this category do not fall into other categories.

From the table, we can observe that all of the works use both direct and indirect trust - as stated by the category. Regarding the Trust Update procedure, most of the papers are using the Event-Driven. More specifically, 57.14% are Event-Driven, 14.3% are using the Time-Driven approach and the rest do not refer to the Trust Update Procedure. All papers present experiments, and 71.4% present the simulator used for the experiments. Moreover, 28.6% preferred a Centralized approach. Regarding Trust Formation, most of the papers present a Multi Trust approach (85.7%) and the rest of the papers provide a Single Trust approach. Finally, 14.3% of the papers define the Threat Model. In the rest of this part, we are going to present a summary of each work.

Aldawsari and Artoli [4] proposed a cluster-based system. They also incorporated a base station (BS) with unlimited energy into the network. For evaluating trust at the cluster level the direct trust of the Cluster-Head and the recommendations of its neighbors are encountered. For trust between two different clusters the cluster heads and the BS is participating in the procedure. The energy consumption is taken into account to calculate the trust. Experiments were conducted on the NS-3 simulator to test the detection rate, energy consumption, and trust evaluation time and to compare the proposed scheme with other methods. The results were presented in diagrams. The authors do not specify the threat model, which is a limitation. Energy consumption must be taken into account in a resource-constrained environment.

Qureshi et al. [64] proposed a trust management system for edge-based IoT networks. The proposed model combines direct and indirect trust to derive the trust level. The system's threat model includes BMA, DoS, and OOA. The trust calculation procedure takes into account the packet drop rate and the packet data rate. OMNET++ used for the following experiments: level of trustworthiness, detection rate, detection accuracy, detection of false positive rate, the impact of a network lifetime, the impact of average packet delay, the impact of average throughput, and end-to-end delay analysis. As a limitation, we state that the threat model can be expanded, so the method will be able to identify more attacks. The authors are considering an edge-based IoT architecture, which is important since IoT and edge devices are collaborating on multiple concepts.

Din et al. [30] proposed a mechanism consisting of IoT-edge nodes, an application programming interface, and a centralized trust agent. The trust agent evaluates the trust level. Based on their trust level, the nodes are allowed to communicate with other nodes. The trust properties used are the following: compatibility, cooperativeness, delivery ratio, and recommendations. The Contiki Cooja simulator was used to acquire the results, the Java language was used for the interaction, and a virtual machine was a platform for simulations. The experiments focused on testing the quality of service (QoS) and resilience against the BMA, BSA, WHA, and SPA. Even though the authors are presenting

TABLE 6. Overview of approaches proposing direct trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[7]	Direct		✓		Multi-Trust	✓	✓	• Does not specify the trusted entities • Deals only with one trust-related attack.	• Messaging system for identifying tampered messages • Isolated untrusted nodes
[28]	Direct	Event-Driven	✓		Multi-Trust	✓	✓	• Neighbor sensors have to gather the same category of data.	• Customized blockchain for IoT
[50]	Direct	Event-Driven	✓	✓	Multi-Trust	✓		• Do not deal with trust-related attacks	• Considering mixed-attacks
[75]	Direct		✓		Multi-Trust		✓	• Do not specify the threat model	• Travel strategy for energy saving • Mobility
[68]	Direct	Event-Driven	✓	✓	Multi-Trust			• Do not specify the threat model	• Takes into account the context
[44]	Direct	Time-Driven	✓		Multi-Trust	✓	✓	• Deals with only 2 kind of attacks	• Takes into account the energy consumption
[72]	Direct		✓		Multi-Trust		✓	• Do not specify the threat model • Do not give details about the deep learning	• Predict the physical attacks and cyber-attacks • Focused on energy consumption
[42]	Direct		✓		Multi-Trust			• Do not specify the threat model	• Metric that provides a perception of a node before interaction

experiments for specific attacks, in the main part of the paper they do not specify the threat model.

Yu et al. [79] proposed a system consisting of nodes and a base station (BS). The trust parameters for direct trust are the packet forwarding capacity, the repetition rate, the consistency of the packet content, the delay, the integrity, etc. For the calculation of indirect trust, the D-S theory was used. Experiments were conducted on MATAB to test the performance of the solution and its behavior under different attacks and compare the method to other schemes. Also, energy consumption was tested, a crucial parameter in a resource-constrained environment. The authors do not specify the threat model.

Mendoza and Kleinschmidt [55] proposed a distributed trust management model for multi-service IoT using direct and indirect observations. The trust management scheme assigns positive scores for honest nodes and negative scores for malicious nodes, using direct interactions between nodes (service requests) and recommendations from neighbors (by exchanging trust tables). The authors implemented malicious nodes performing the BMA to analyze the effectiveness of the model. The obtained results from the experiments conducted on the Cooja simulator show the proposed trust management model detects malicious behavior in the network considering topologies with 10% to 30% of malicious nodes. This model may be used to detect other common attacks in the IoT. The authors do not specify the threat model.

Awan et al. [12] proposed a mechanism that mainly works with direct observations and asks for recommendations if no interactions were recorded in the past. The authors also took into account scalability, since the nodes only store the results of the experience component. Other trust parameters are reputation and knowledge. The following experiments took place on the NS-3 simulator: the behavior of the solution was tested on BSA, BMA, and OOA; comparison with other models; and energy consumption. Even though the authors are conducting experiments under different attacks, they do not specify the threat model in the main part of the paper or the architecture of the network. The nodes are storing one component of trust for scalability and storage reasons and the authors are taking into account the energy consumption.

Din et al. [29] are considering a resource-sharing environment consisting of resource providers and resource seekers. The network also has a central authority that coordinates the procedures. The resource providers are nodes that want to share their resources, and the resource seekers need more resources. A resource is allocated to a specific node for a specific amount of time. The resource providers have to make an offer to the interface, and the resource seekers check the availability and choose the offer that best fits their needs. When a resource seeker wants to use some resources from the resource provider, the trust evaluation procedure is triggered using previous observations. The trust evaluation consists of two components: competence, which involves stability and cooperativeness, and trustworthiness, which includes persistence and reputation. During the first interaction, the procedure takes into account only recommendations. The final trust values are compared to a threshold. The authors do not specify the threat model, but they present a framework where resource-constrained nodes can share and exploit free resources from other nodes in the network.

C. FUZZY LOGIC

Boolean logic permits expressions that are either true or false. However, in real life sometimes truth is a spectrum. Fuzzy logic is a multi-value logic that permits intermediate values between true and false [22], [39]. Some works on trust management exploit the nature of fuzzy logic to represent trust as a spectrum between trusted and not trusted. A summary of the findings is presented in Table 8.

We can observe from the table that all of the papers provide experiments. We can see that 80% of the papers mention the simulator used to conduct the experiments. Regarding the Trust Update, 60% of the paper preferred an Event-Driven approach, while the rest (40%) the Time-Driven method. It is interesting to notice that none of the papers use a Centralized authority for trust evaluation, and none of the papers prefer a Single Trust approach. It is also worth pointing out that 20% of the papers in this category do not use direct or indirect trust. Finally, 40% of the papers define the Threat Model.

Alshehri et al. [7] proposed a cluster-based architecture, including one super node and many main nodes that are responsible for multiple cluster nodes. Only the cluster nodes are considered malicious. The malicious nodes can perform OOA. Alshehri et al. introduced five algorithms to calculate the trust score of every cluster node. To calculate the trust score, they take into account the quality of service, the

TABLE 7. Overview of approaches proposing trust management methods using recommendations.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[4]	Both	Time-Driven	✓		Single-Trust		✓	• Do not specify the threat model	• Takes into account energy consumption
[64]	Both	Event-Driven	✓		Multi-Trust	✓	✓	• Limited attacks in threat model	• Edge-based IoT
[30]	Both	Event-Driven	✓	✓	Multi-Trust			• Do not specify the threat model	•
[79]	Both		✓		Multi-Trust		✓	• Do not specify the threat model	• Energy consumption was tested
[55]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	•
[12]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model • Do not specify the architecture of the network	• The nodes only store one component of trust for scalability • Takes into account energy consumption
[29]	Both		✓	✓	Multi-Trust			• Do not specify the threat model	• Focused on resource sharing

history score, and the trust score. They use a fuzzy approach to classify the trust scores into fuzzy sets. The nodes are then classified into three categories: trusted, semi-trusted, and non-trusted. Based on the trust score, the nodes can change clusters, and based on the category, they can perform specific acts. They performed experiments in the Cooja simulator regarding scalability, the accuracy of different attacks, and fuzzy and non-fuzzy approaches, and they presented diagrams with the results. The authors do not specify the trusted entities involved in the procedure. Also, they only deal with one trust-related attack. Finally, they proposed a HEXA decimal-based messaging system that can be used to detect tampered messages in transit, and they isolate the untrusted nodes from the network.

Guleng et al. [37] proposed a trust management architecture for vehicular ad hoc networks (VANET). Consequently, we can conclude that they studied dynamic topology. The proposed scheme is distributed and uses fuzzy logic to evaluate direct trust. For indirect trust, the authors proposed a reinforcement learning approach. To calculate the trust scores, they take into account the cooperativeness, honesty, responsibility factor, and previous values. Also, the proposed trust management takes into account the trust in a message. They performed experiments using the NS-2.34 simulator to compare their method with ''w/o Trust'' and ''Deterministic Trust''. Also, a simulation of BMA was performed. The final results of the experiments were presented in diagrams. The proposed solution does not specify the threat model. It is a nice approach that the authors are taking into account both the node's trust and the message's trust. The solution can also be applied to a high-mobility network.

Mahmud et al. [53] proposed a trust management method for cloud-based architecture for neuroscience applications. The proposed method estimates the trust level using an adaptive neuro-fuzzy inference system (ANFIS) and weighted additive methods. Furthermore, it is worth pointing out that this technique takes into account the behavior of the node and the trustworthiness of the generated data. Behavioral trust takes into account: the Relative Frequency of Interaction, intimacy, honesty, previous interactions, and indirect trust. Data trust depends on the deviation of a node's instantaneous data from its historical data and indirect recommendations. NS-2 simulator was used to perform the following experiments: Packet Forwarding Ratio, Network Throughput, Average Energy Consumption Ratio, Accuracy, F-measure, Comparison with other models, and different linguistic terms (5 and 3). The final results of the experiments were presented in diagrams. The authors do not specify the threat model. In our opinion, it is a pro that they take into account data's trust and energy consumption.

Chen et al. [23] focuses on a dynamic architecture, which means that some nodes may leave or enter the network. The nodes are divided into Service Providers and Service Requestors. This work proposes a distributed fuzzy logic trust management scheme. The model consists of both direct trust (monitoring the neighbors) and indirect trust (recommendations). To calculate the trust, the following values are considered: the end-to-end forwarding ratio (EPFR), the average energy consumption (AEC), and the packet delivery ratio(PDR). Also, a global trust can be issued to obtain a more accurate value of trust. For the experiments, the NS-3 simulator was used, and the following values were taken into account: EPFR, AEC, PDR, convergence speed, detection probability, and comparison with other models. The final results of the experiments were presented in diagrams. There is a need for more efficient global trust computations.

Esposito et al. [33] proposed a system consisting of IoT nodes that communicate with edge nodes that participate in a blockchain. The blockchain stores a smart contract that periodically stores the trust scores. The smart contract receives the real numbers extracted from the nodes and uses fuzzy logic to translate the real numbers into linguistic terms. The main threat to the system is to store on the blockchain a false value for the computed trust score. So, the solution is to focus on finding a way to reject these kinds of false messages. A game-theoretic approach was employed, forming a game between the edge node and a common node. A node has two available actions: to send a message or to avoid sending a message. This message might contain malicious data. At the reception of the message, the edge node can do only two possible actions: Y indicating that it accepts the message and passes it to the blockchain participants to update its state, or N indicating that it rejects the message and does not pass it to the blockchain. Real sensors were used for performing the following experiments: comparison with other models, belief evolution, and attack success probability with and without the proposed defense. The final results of the experiments were presented in diagrams. The authors do not deal with blockchain-related attacks. Messages coming from untrusted sources are blocked.

TABLE 8. Overview of approaches proposing fuzzy logic trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[7]	Direct		✓		Multi-Trust	✓	✓	• Does not specify the trusted entities • Only one trust-related attack	• Messaging system for identifying tampered messages • Isolated untrusted nodes
[37]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account both the trust of the node and the message • All nodes can be malicious • Takes into account the mobility
[53]	Both	Both	✓		Multi-Trust		✓	• Do not clearly specify the trust model	• Takes into account the data trust • Takes into account Energy consumption
[23]	Both	Event-Driven	✓		Multi-Trust		✓	• Need for more efficient global trust computation • Do not specify the threat model	• Takes into account Energy consumption
[33]		Time-Driven	✓			✓		• Do not study blockchain-related attacks	• Blocks messages from untrusted nodes.

D. BLOCKCHAIN BASED

The blockchain is a distributed database or ledger first introduced in [57] as an underlying technology for Bitcoin. Blockchain provides immutability since all the participants are managing the chain through a consensus mechanism. This property made the blockchain popular in different applications. In the scope of trust management for IoT, some works exploit the properties of the blockchain to calculate or store trust data. A summary of the results can be seen in Table 9.

We can observe from the table that all of the papers provide experiments, but only 37.5% of the papers provide the simulator used. None of the papers proposes a Centralized approach, which makes sense since we are examining blockchain-based methods. Regarding information gathering: 12.5% are using only direct trust, 50% are using recommendations, and 37.5% do not use an information gathering technique. Most of the approaches are Event-Driven (62.5%), while the rest (37.5%) are Time-Driven. In total, 87.5% are referring to the trust formation, with 71.4% using the Multi-Trust scheme. Finally, 75% defines the Threat Model.

Dedeoglu et al. [28] proposed a system containing sensors and gateways. Gateways run the blockchain and are associated with several sensors. A malicious sensor can tamper with the data, and a malicious gateway can generate invalid blocks. A lightweight block generation scheme was proposed where blocks are generated at time intervals. The block validation mechanism adapts the block validation scheme based on the reputation of the node that generated the block and the number of validators. For the consensus mechanism, the following method was introduced: If a validator detects an invalid transaction, it broadcasts INVALID and the nodes have to validate the transaction; otherwise, the block is appended to the blockchain. The proposed technique evaluates the trustworthiness of sensor observations. The sensor assigns a confident value to the data and sends it to a gateway. The gateway compares the data with the data of the other cluster sensors (assumption: the sensors in the same cluster have correlated data). In the end, based on the result, the reputation of the node is recalculated. The gateways store the information on a blockchain. The reputation of the gateways is calculated based on their actions during the generation and validation of the blocks. The trust parameters taken into account are the confidence of the data source, the reputation of the data source, and evidence from other observations. The experiments took place in the NS-3 simulator, and they are both blockchain-related and trust-related. The final results were presented in diagrams. One limitation of this approach is that the neighboring sensors have to gather the same category of data; otherwise, it cannot be applied.

Debe et al. [27] proposed a scheme based on the Ethereum blockchain where gateways are presented as full nodes of the blockchain and sensors as lightweight nodes. They propose a data attestation solution. The lightweight nodes get some responses from the full nodes. These responses are validated by other full nodes. The attack this work tackles is EA. The trust is calculated on a smart contract, and the parameter used to assess the trustworthiness of a gateway is the client's feedback. The code is publicly available, and the experiments focused on smart contract code vulnerability analysis with tools. This approach stores one trust score per node. If a node is behaving maliciously only toward a small set of nodes, its trust score will still be high. The authors performed cost analysis in terms of gas.

Putra et al. [63] proposed a system where sensors are divided into Service Providers and Service Consumers, which consist of the lightweight clients in the main blockchain. Also, a set of permissioned blockchains are implemented to maintain the sensitive data of the sensors. These chains are maintained by a consortium of independent and partially trusted entities. The following attacks can be performed by a malicious node: BMA, RA, PA, BSA, WA, and SA. The trust is calculated on a smart contract on the main blockchain. The experiments took place in a real environment. For the blockchains, the Rinkeby Ethereum test network was used as the main blockchain, and private chains were used for storing sensitive data. The following experiments took place: different kinds of weights for calculating the trust scores, comparison with other schemes trust and reputation convergence, latency, and required gas. The final results were presented in diagrams and tables. This approach requires a cluster of partially trusted computers to run the private blockchain. It might be optimistic for some cases to hold such a cluster. On the other hand, this approach must separate sensitive from publicly available data, preserving privacy.

Kouicem et al. [46] proposed a system based on a fog architecture that consists of the following components: IoT

service requesters, Service providers, and fog nodes that are responsible for trust management. The Service providers and the nodes participate in the blockchain. The paper proposes a new consensus mechanism. Each IoT object can assess the trustworthiness of a service provider and share it. Exploiting the blockchain architecture, this protocol provides a global image of trust values. The malicious service providers can perform BMA, SPA, BSA, OOA, and OSA. The malicious fog nodes can drop, delay, modify, and redirect the received messages. The following parameters are taken into account for the trust evaluation procedure: A set of criteria reported on the blockchain for direct trust, previous interactions, and recommendations. For the experiments, a private blockchain was used, and the consensus mechanism was a combination of PBFT and PoS. The following experiments were performed: different kinds of weights, blockchain scalability evaluation, and comparison with other schemes. The results were presented in diagrams. The authors are dealing with only trust-related attacks and make assumptions about the security of the blockchain. However, they deal with a high-mobility environment and propose a new consensus mechanism.

Bordel et al. [20] proposed a method where a trusted third party acts as the TTP. The IoT messages are controlled by a third party. This party is acting between the IoT nodes and the blockchain. The architecture is based on Blockchain technology and the computation of different conceptual models (cognitive, computational, neurological, and game theoretical) using stochastic functions. Smart contracts are employed to calculate global trust. Matlab 2020 was used to perform the following experiments: convergence time and success rate. The results were presented in diagrams. The authors do not specify the threat model or give details about the blockchain technology used. However, they propose multiple concepts of trust.

Amiri-Zarandi et al. [9] proposed a fog architecture scheme. The interactions between the nodes and the blockchain can occur directly or via edge nodes. The nodes are divided into clusters, and they communicate with a fog device. The social connection between the devices is used for trust evaluation. The scheme also works with recommendations that are filtered by a lightweight algorithm. The blockchain is used for storing trust-related data. The malicious nodes can perform BMA, BSA, SFA, and DoS. Honesty was used as the trust parameter. Also, the following experiments took place using the Ethereum blockchain: performance evaluation and experimental comparison with other models. The results were presented in diagrams and tables. The authors do not study any blockchain-related attacks. This approach dynamically selects counselors, as one node might be a good fit at first but behaves maliciously afterward. Also, they perform cost analysis in terms of gas.

Esposito et al. [33] proposed a system consisting of IoT nodes that communicate with edge nodes that participate in a blockchain. The blockchain stores a smart contract that periodically stores the trust scores. The smart contract receives the real numbers extracted from the nodes and uses fuzzy logic to translate the real numbers into linguistic terms. The main threat to the system is to store on the blockchain a false value for the computed trust score. So, the solution is to focus on finding a way to reject these kinds of false messages. A game-theoretic approach was employed, forming a game between the edge node and a common node. A node has two available actions: to send a message or to avoid sending a message. This message might contain malicious data. At the reception of the message, the edge node can do only two possible actions: Y indicating that it accepts the message and passes it to the blockchain participants to update its state, or N indicating that it rejects the message and does not pass it to the blockchain. Real sensors were used for performing the following experiments: comparison with other models, belief evolution, and attack success probability with and without the proposed defense. The final results of the experiments were presented in diagrams. The authors do not deal with blockchain-related attacks. Messages coming from untrusted sources are blocked.

Jeribi et al. [43] proposed a solution that includes a network of smart buildings where a lot of IoT devices are installed. There is also a verification manager making access decisions. Each IoT device is connected to a trust management system which is responsible for evaluating the trust of other nodes and producing a complete trust level. The technique takes into account both direct and indirect trust, for direct trust computations, cooperativeness, knowledge, and a group of interest are taken into account. After trust computation, a machine learning algorithm is deployed to classify the trust and determine the most trustworthy device. For this purpose, the ID3 algorithm was used. This algorithm is a supervised learning technique that chooses the best feature that produces the lowest amount of entropy. As an input, it takes an array of trust values and the output is a decision tree of nodes based on the trust values. The trust value of the root node serves as the threshold. Afterward, the values are sent to the blockchain, where they are stored. A permission-based private blockchain was employed. When a new node enters the network, the trust manager calculates the trust value. If it is above the threshold value, it is passed to the blockchain, where it validates that the trust value matches the threshold and is stored in the trustworthy devices. The procedure is repeated at fixed intervals. The method was tested in BSA, BMA, and OOA. This approach requires a network of trusted nodes to run the blockchain, which is not always applicable in a real-life setting. Also, they do not specify the threat model, even though they perform experiments on specific attacks. This approach deals with the cold-start problem, which is a crucial issue in dynamic environments.

E. GAME THEORY

Game theory provides the framework to describe the strategic interaction between rational players. The nodes of a system can be seen as rational players since they try to maintain a high trust score to be selected as service providers. Therefore, some works model the trust management method as a game

TABLE 9. Overview of approaches proposing blockchain-based trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[28]	Direct	Event-Driven	✓		Multi-Trust	✓	✓	• Neighbor sensors have to gather the same category of data.	• Customized blockchain for IoT
[27]		Event-Driven	✓		Single-Trust	✓		• One score per node, might be malice to specific nodes • Deals only with one attack	• Available Code • Cost analysis in terms of gas
[63]	Both	Event-Driven	✓		Multi-Trust	✓		• Many partially trusted computers to run the private chains • Issue with a new node with 0 score • Only trust-related attacks	• Separation between sensitive and publicly available data.
[46]	Both	Time-Driven	✓		Multi-Trust	✓		• Make assumptions about blockchain-related attacks • Only trust-related attacks	• Mobility • New consensus method
[20]		Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model • Do not give details about the blockchain technology used	• Multiple trust concepts
[9]	Both	Event-Driven	✓		Single-Trust	✓		• Do not study blockchain-related attacks	• Dynamically selected counselors • Cost analysis in terms of gas
[33]		Time-Driven	✓			✓		• Do not study blockchain-related attacks	• Blocks messages from untrusted nodes.
[43]	Both	Time-Driven	✓		Multi-Trust		✓	• A trusted network of nodes is required to run the blockchain • Focused on smart buildings Do not specify the threat model	• Deals with the cold-start problem.

using game theoretic approaches. A summary of the results can be seen in Table 10.

We can observe from the Table that all of the papers present experiments and define the Threat Model. Even though all the papers provide experiments, 25% of them state the simulator used. Also, all of the papers provide a Time-Driven solution, but 25% of them provide a hybrid scheme with both an Event and Time driven approach. None of the papers involve a centralized authority in their system. Regarding the Trust Formation: 25% of them are use multiple trust parameters, 25% only a single trust parameter, and 50% do not specify the trust formation.

Djedjig et al. [31] proposed a distributed cooperation-trust-based routing mechanism for RPL, where the malicious nodes can perform rank attacks and BA. At each hop of an RPL routing path, the child node selects the node that has a higher trust value, more energy, and better link quality as its preferred parent. The trust is calculated by taking into account energy consumption, honesty, selfishness, and the ETX. Also, they translated the proposed trust management method into a strategy using game theory concepts. A non-trusted node will be discarded from the network. So, there is no advantage for a rational player to misbehave since it will be discarded from the network. The foundation of the solution is a non-zero-sum, non-cooperative iterated PD game. Experiments were performed using the Cooja simulator and were focused on comparing the proposed method with other schemes in terms of throughput, energy, Average Node Rank Changes under Blackhole and Rank attacks, and Average Packet Delivery Ratio. The results were presented as diagrams. The authors are focused on routing security and only deal with a set of trust-related attacks. They also test energy consumption, which is positive in a resource-constrained environment.

Esposito et al. [33] proposed a system consisting of IoT nodes that communicate with edge nodes that participate in a blockchain. The blockchain stores a smart contract that periodically stores the trust scores. The smart contract receives the real numbers extracted from the nodes and uses fuzzy logic to translate the real numbers into linguistic terms. The main threat to the system is to store on the blockchain a false value for the computed trust score. So, the solution is to focus on finding a way to reject these kinds of false messages. A game-theoretic approach was employed, forming a game between the edge node and a common node. A node has two available actions: to send a message or to avoid sending a message. This message might contain malicious data. At the reception of the message, the edge node can do only two possible actions: Y indicating that it accepts the message and passes it to the blockchain participants to update its state, or N indicating that it rejects the message and does not pass it to the blockchain. Real sensors were used for performing the following experiments: comparison with other models, belief evolution, and attack success probability with and without the proposed defense. The final results of the experiments were presented in diagrams. The authors do not deal with blockchain-related attacks. Messages coming from untrusted sources are blocked.

Rani et al. [66] considered several sensor nodes, deployed randomly in a network field. All these nodes are equipped with limited-power batteries and have a short radio range. A base station with an unlimited source of energy as a central administrative authority is also deployed in the network field. It is also considered that the nodes of the network form clusters. A cluster consists of cluster members and a cluster head. The proposed scheme uses evolutionary game theory in cluster formation and non-cooperative game theory to detect malicious nodes in the network. When a node receives a trust request, it has two possible actions: to reply or not reply. When a node replies, it has some communication cost, which helps in energy efficiency. The malicious nodes can perform BMA, OOA, packet modification, collusion attacks, DoS, BA, and WHA. The experiments performed on the NS-3 simulator focused on: detection rate, average energy consumption, comparison with other schemes, trust evaluation time, and detection time. The results were presented in diagrams. The authors are presenting a cluster formation solution, which is a main issue in cluster-based networks. One drawback we identified is that the authors can expand their solution to cover more attacks.

Duan et al. [32] adopted watchdog. Each sensor node is responsible for monitoring the behavior of its neighbors. A WSN was considered to consist of a few sink nodes and several sensor nodes. The main goal of this paper is to reduce energy consumption and latency for trust

TABLE 10. Overview of approaches proposing game theoretic trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[31]	Both	Both	✓		Multi-Trust	✓	✓	• Focus only on routing security. • Deals only with a set of trust-related attacks	• Different analysis for the game theoretic approach • Tests energy consumption
[33]		Time-Driven	✓			✓		• Do not study blockchain-related attacks	• Blocks messages from untrusted nodes.
[66]	Both	Time-Driven	✓			✓	✓	• Expand the threat model	• Cluster formation solution
[32]	Both	Time-Driven	✓		Single-Trust	✓		• Overhead produced by trust request	• Really focused on energy consumption

evaluation. The paper proposed a method to find the optimal number of recommendations needed for trust evaluation while maintaining a high-security level. The nodes are considered players with the following strategies: reply or not reply to save energy for trust computation. So this paper is proposing a dilemma game. The malicious nodes may perform BMA, DoS, or selfish attacks. The parameter used to calculate the trust is energy. NS-2 simulator was used to perform the following experiments: optimal selection of some values related to the trust process and comparison with other mechanisms. The results are presented in diagrams. One drawback of the solution is the overhead produced by the trust requests. On the other hand, the authors are examining energy consumption.

F. CONTEXT

One node can participate in many contexts and behave differently in each one. Some works take into account the different contexts to evaluate the trustworthiness of the node. The summary of the findings for this category is presented in Table 11.

We can observe from the table that all of the papers use an Event-Driven approach, provide experiments, and use multiple parameters for trust calculation. 16.6% of the papers use only direct observations for information gathering, while the rest use recommendations. Also, 16.6% of the papers use a central authority for deriving trust. Even though all the papers provide experiments, only 33.34% of them indicate the simulator used. Finally, 16.6% of the papers define the threat model.

Rafey et al. [65] proposed a system where nodes form communities of interest. The proposed model takes social relationships into account to evaluate trust. Also, the trust is calculated in the different contexts in which the node is participating, and the final trust is the sum of the individual ones. This work also presents a way for storing trust values. The trust is derived from node transaction factors: Computational power, Context importance, Confidence, feedback, and social relationship factors: owner trust and SIoT relationship. The malicious entities can be: individual malevolent nodes, malevolent collectives, malevolent spies, malevolent pre-trusted nodes, partially malevolent collectives, or malevolent collectives with camouflage, and they can perform SA and BMA. The experiments were focused on performance and comparison with other mechanisms. The final results were presented in diagrams. The authors are dealing only with trust-related attacks. On the other hand, they propose a trust storage mechanism, which is helpful for resource-constrained devices. Moreover, they include social aspects in their solution.

Altaf et al. [8] proposed a system consisting of users and service providers. The users are requesting services from the service providers. The context was used to calculate trust in a different context. One server has different trust scores for every context. Each edge node takes recommendations from context-similar nodes to calculate the trust of serving nodes. The trust parameters used are the following: server capability in terms of service provided, location, type of server, Quality of service, similarity with the recommender, location of the servers, and list of requested services. The experiments were focused on performance, resilience, and comparison with other models. The final results were presented in diagrams. The authors are filtering out the recommendations based on the context. Sometimes, a node might behave differently in different contexts. However, the authors do not specify the threat model.

Saied et al. [68] proposed a context-aware and multi-service trust management system. Upon a request from a node asking for assistance, the trust manager starts the entity selection process to return a set of trustworthy assisting nodes to the requester. A set of recommenders sends reports; the most important are those that lie to the same or more similar services and recent ones. A quality of recommendation score is assigned to each node, reflecting its trustworthiness when rating other nodes. The context was used to filter out recommendations and select the most relevant ones. The trust is calculated based on the following parameters: The score is given by the requester node to the service provider evaluating the offered service, a weight that depends on time, similarity, and quality of recommendations. Experiments were performed focused on the comparison of reactions against different kinds of attacks like on-off, bad-mouthing, and selective behavior attacks. The authors are filtering out the recommendations based on the context. Sometimes, a node might behave differently in different contexts. However, the authors do not specify the threat model, even though they are performing experiments on different attacks.

Abidi and Azzouna [2] proposed a system consisting of nodes that create social relationships, a context Manager, a social relationship manager, and a trust formation adjustor. The goal of the system is to assist the nodes in finding trustworthy service providers. The level of trust between the service providers depends on both direct trust (the

interactions between the requestor and the service provider) and, recommendations of the requestor's neighbors. The trust parameters adjust to the network context and the relationships between the nodes. The trust parameters taken into account are the quality of Service and social trust properties like honesty, cooperativeness, and social relationships. Trust calculations rely on Social relationship factors. The following are the social relationships that are formed in the model: Parental Object Relationship, co-location Object Relationship, co-work Object Relationship, ownership Object Relationship, and social object relationships. The experiments were performed in MATLAB and they were focused on the performance of the method and comparison with other models. The final results were presented in diagrams. The authors do not specify the threat model, but they take social aspects into account.

Adewuyi et al. [3] proposed a system called CTRUST. In this work, the authors model the trust units with mathematical functions. The trust properties used to calculate the trust level are the social relationships between the nodes and the context. This paper also introduces a parameter to model trust maturity, the point at which trust can be computed using direct interactions alone. The performance was evaluated based on trust accuracy, convergence, and resiliency. Diagrams present the final results. The authors do not specify the threat model, but they take social aspects into account.

Magdich et al. [52] are considering an environment consisting of a set of users and a set of devices. A user can own one or multiple devices, and the devices can provide or request one or more services. Every device is represented by a vector containing three values: user, device, and environment (public or private). The environment value sets the threshold of trust. Also, each device stores its characteristics (manufacturer, type, capacity, and location), the profile of its owners (friendship, CoI, and Co-work), the transaction history between other nodes, and trust values. In this approach, the trust of the owner, the device, and the environment are taken into account to decide the trust value. The method also uses recommendations. Afterward, the threshold of trust has to be decided. The authors are proposing a Machine Learning technique that they compare with the static method, available in the literature. The ML (Artificial Neural Network) algorithm classifies the nodes as trustworthy or not. After each interaction, the nodes evaluate each other and share the result with the other nodes as a recommendation. The authors do not specify the threat model, but they take social aspects into account.

G. SOCIAL

Social IoT provides a combination of IoT and social networking. The sensors can establish social relationships [56]. Some works exploit this aspect to create trust management methods. A summary of the findings for this category can be seen in Table 12.

We can observe from the table that 95.6% of the papers use both direct and indirect trust, and only 4.4% use only direct observations. Regarding the Trust Update 65.2% of the papers use the Event-Driven approach, 13% use the Time-Driven approach, and 13% do not refer to the Trust Update. It is worth pointing out that 4.4% of the papers use a hybrid approach to the Trust Update. Also, 95.6% are presenting Experiments but only 30.4% of them are referring to the simulator used. 8.7% of the papers preferred a Centralized authority to manage the trust calculation procedure. Regarding Trust Formation: 8.7% of the papers are using the Single-Trust approach, while the rest are using the Multi-Trust approach. Finally, 52.2% of the papers define the Threat Model.

Rafey et al. [65] proposed a system where nodes form communities of interest. The proposed model takes social relationships into account to evaluate trust. Also, the trust is calculated in the different contexts in which the node is participating, and the final trust is the sum of the individual ones. This work also presents a way for storing trust values. The trust is derived from node transaction factors: Computational power, Context importance, Confidence, feedback, and social relationship factors: owner trust and SIoT relationship. The malicious entities can be: individual malevolent nodes, malevolent collectives, malevolent spies, malevolent pre-trusted nodes, partially malevolent collectives, or malevolent collectives with camouflage, and they can perform SA and BMA. The experiments were focused on performance and comparison with other mechanisms. The final results were presented in diagrams. The authors are dealing only with trust-related attacks. On the other hand, they propose a trust storage mechanism, which is helpful for resource-constrained devices. Moreover, they include social aspects in their solution.

Nitti et al. [59] proposed a system consisting of a network of nodes, several pre-trusted entities to hold a distributed hash table structure, and four other components to manage the network: relationship management, service discovery, service composition, and trustworthiness management. This paper defined two models for trustworthiness management: subjective and objective. In the first model, each node computes the trustworthiness of its friends using its own experience and the opinions of the friends in common. In the second model, the information about each node is stored on a distributed hash table structure. The following trust parameters are taken into account: feedback, the total number of transactions, credibility, the transaction factor, the relationship factor, the notion of centrality, and computation capability. Trust calculations rely on Social relationship factors. The following are the social relationships that are formed in the model: Parental Object Relationship, co-location Object Relationship, co-work Object Relationship, ownership Object Relationship, and social object relationship. The experiments conducted focus on comparing the performance with other models and how the proposed approaches work with three different dynamic behaviors of the nodes. This approach requires some pre-trusted entities to

TABLE 11. Overview of approaches proposing context based trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[65]	Both	Event-Driven	✓		Multi-Trust	✓		• Only trust-related attacks	• Proposal for trust storage • Takes into account social aspects
[8]	Both	Event-Driven	✓		Multi-Trust			• Do not specify the threat model	• Filters out dissimilar recommendations
[68]	Direct	Event-Driven	✓	✓	Multi-Trust			• Do not specify the threat model	• Filters out dissimilar recommendations
[2]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account social aspects
[3]	Both	Event-Driven	✓		Multi-Trust			• Do not specify the threat model	• Takes into account social aspects
[52]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account social aspects

be involved, which is not always applicable in real life. Also, they do not provide the threat model. Finally, a pro is that they provide a view of trust of the whole system as a holistic evaluation.

Mon et al. [10] proposed a cluster-based system with a central trust entity. Initially, a cluster is formed, and the main node is selected based on its trust scores, which include QoS and Social trust properties. The main node is periodically updated based on the trust value using regression model-based clustering. Experiments were conducted to test the performance of the proposed approach. The results were presented in diagrams. One limitation is that trust values are required to form clusters; it is not clear what happens during bootstrapping. One strength of the solution is that it takes into account data trust, which is important for identifying false messages even from trustworthy nodes.

Marche and Nitti [54] proposed a system that focuses on detecting trust attacks. To achieve this, machine learning techniques are applied. Trust calculations rely on Social relationship factors. The following are the social relationships that are formed in the model: Parental Object Relationship, co-location Object Relationship, co-work Object Relationship, ownership Object Relationship, and social object relationship. A malicious node can be malicious to everyone or selectively and perform the following attacks: OOA, WHA, BMA, BSA, SA, and OSA. For trust computations, the following parameters are taken into account: previous interactions, computation capabilities, relationship factors, external opinions, and dynamic knowledge. Experiments were conducted to test the performance of the iSVM. Diagrams present the results of the experiments. One limitation is that the approach only deals with trust-related attacks. On the other hand, at the steady state, only one parameter is required to compute the trust value, which saves time and energy.

Abidi and Azzouna [2] proposed a system consisting of nodes that create social relationships, a context Manager, a social relationship manager, and a trust formation adjustor. The goal of the system is to assist the nodes in finding trustworthy service providers. The level of trust between the service providers depends on both direct trust (the interactions between the requestor and the service provider) and, recommendations of the requestor's neighbors. The trust parameters adjust to the network context and the relationships between the nodes. The trust parameters taken into account are the quality of Service and social trust properties like honesty, cooperativeness, and social relationships. Trust calculations rely on Social relationship factors. The following are the social relationships that are formed in the model: Parental Object Relationship, co-location Object Relationship, co-work Object Relationship, ownership Object Relationship, and social object relationships. The experiments were performed in MATLAB and they were focused on the performance of the method and comparison with other models. The final results were presented in diagrams. The authors do not specify the threat model, but they take into account the context which is a strength since nodes might act differently in different contexts.

Nitti et al. [58] proposed a system where the nodes evaluate the trustworthiness of other nodes based on their observations and recommendations of common friends (between the trustor and the trustee). The trust parameters taken into account are feedback, the total number of transactions, the credibility, the transaction factor, the relationship factor, the notion of centrality, and computation capability. Trust calculations rely on Social relationship factors. The following are the social relationships that are formed in the model: Parental Object Relationship, co-location Object Relationship, co-work Object Relationship, ownership Object Relationship, and social object relationship. Experiments were conducted to test the performance of the proposed solution. The results are presented in diagrams. The limitation of this solution is that the authors do not specify the threat model.

Wang et al. [74] proposed a trust model based on direct and indirect trust computation with trust prediction. The prediction method depends on the combination of exponential smoothing and a Markov chain. Exponential smoothing was employed to predict trust and a Markov chain was employed to fix any deviation. Thus, a prediction method was employed to predict the current trust level based on interaction history, behavior history, and some other factors like the device model. For the trust computation, both social and unsocial parameters were considered. The following experiments were performed: comparison of trust prediction with different exponential smoothing coefficients; comparison between first and second exponential smoothing; and experiments for different kinds of attacks. The results were presented as diagrams. The authors do not specify the threat model,

which is a drawback. On the other hand, they are proposing a solution to the communication latency issue.

Chen et al. [25] designed and analyzed a trust management protocol for SOA-based IoT systems. The IoT owners can share their feedback, so a filtering method was proposed to select the feedback of owners with common interests. Also, the nodes can adjust the weights of direct trust and recommendations. The social aspects were used to weigh the recommendations: Friendship, social contact, and community of interest. User satisfaction is the trust parameter used to calculate trust. They also introduced a method for trust storage. The malicious nodes can perform BMA, SPA, BSA, and OSA. The experiments were conducted on the NS-3 simulator, and they tested convergence, accuracy, resiliency, the effectiveness of storage management protocols, and comparative analysis. The results are presented in diagrams. The main limitation of this approach is the threat model, which only includes trust-related attacks. On the other hand, the authors are proposing a trust-based storage solution for resource-constrained devices.

Bao and Chen [16] proposed a dynamic trust management protocol for IoT systems. The trust evaluation takes into account direct and indirect recommendations and social aspects. They also take into account scalability in terms of storing trust values. The malicious nodes can perform BMA, SPA, and BSA. The nodes form communities, and during the trust evaluation procedure, the social aspects considered are honesty, cooperativeness, and the community of interest. Experiments were conducted to observe the effect of some weights used in the trust evaluation and the protocol's resiliency to trust attacks. The results are presented in diagrams. One limitation of this approach is the threat model, which only includes trust-related attacks. On the other hand, the authors are testing the scalability of storing trust values.

Kowshalya and Valarmathi [47] presented a system where the network is presented as a graph. Where V are the participants and E represents the edges between them. The devices form communities of interest based on parental, co-work, and co-location relationships. Also, this paper ensures secure communication among SIoT nodes through simple secret codes. For the trust evaluation procedure, the properties taken into account are the following: honesty, cooperativeness, community of interest, and energy. For the experiments, the SWIM platform was used, and the experiments tested the performance of the proposed model and compared it with other schemes. The results were presented in diagrams. During the experiments, the effect of the trust weights was analyzed. The authors do not specify the threat model, which constitutes a drawback. But on the other hand, they are proposing a way to ensure secure communications using secret codes.

Adewuyi et al. [3] proposed a system called CTRUST. In this work, the authors model the trust units with mathematical functions. The trust properties used to calculate the trust level are the social relationships between the nodes and the context. This paper also introduces a parameter to model trust maturity, the point at which trust can be computed using direct interactions alone. The performance was evaluated based on trust accuracy, convergence, and resiliency. Diagrams present the final results. The authors do not specify the threat model, but they take into account the context.

Bao et al. [18] proposed a system where the nodes form communities of interest. The protocol is distributed and each node evaluates the trust of nodes that share interests. The system can adapt to changes in communities of interest by dynamically selecting the trust parameters. For scalability, the authors also proposed a storage management strategy to save memory from the resource constraint of IoT devices. The malicious nodes can perform BMA, SPA, and BSA. The experiments tested the effect of changing some weight values related to the trust evaluation procedure and the trust evaluation with limited storage space. The results are presented in diagrams. The main limitation of this approach is the threat model, which only includes trust-related attacks. On the other hand, the authors are proposing a trust-based storage solution for resource-constrained devices.

Das et al. [26] proposed a system consisting of IoT nodes and Fog nodes. The IoT nodes communicate with the closest Fog node. This paper proposes a community-based trust management architecture by considering self-trust, social trust, green trust, and QoS trust. Experiments were conducted on MATLAB concerning the performance of the system. Diagrams present the final results of the experiments. The authors do not specify the threat model. On the other side, they are testing the energy consumption of their solution.

Awan et al. [11] proposed a multilevel architecture system. The nodes form communities of interest. Every community has a server to calculate trust. A set of communities forms a domain that has a server to calculate the trust of the domain. The whole system is governed by a server that is responsible for the trust of all the domains. The trust properties taken into account are compatibility, honesty, and competence. The authors are presenting a purely theoretical model in which they do not specify the threat model. On the other side, they are proposing cross-domain trust management, taking into account different domains.

Amiri-Zarandi et al. [9] proposed a fog architecture scheme. The interactions between the nodes and the blockchain can occur directly or via edge nodes. The nodes are divided into clusters, and they communicate with a fog device. The social connection between the devices is used for trust evaluation. The scheme also works with recommendations that are filtered by a lightweight algorithm. The blockchain is used for storing trust-related data. The malicious nodes can perform BMA, BSA, SFA, and DoS. Honesty was used as the trust parameter. Also, the following experiments took place using the Ethereum blockchain: performance evaluation and experimental comparison with other models. The results were presented in diagrams and tables. The authors do not study any blockchain-related

attacks. This approach dynamically selects counselors, as one node might be a good fit at first but behaves maliciously afterward. Also, they perform cost analysis in terms of gas.

Jayasinghe et al. [42] proposed a system where the nodes form communities of interest. In this model, the transactions are under evaluation and should be determined if a transaction is trustworthy. The trust parameters used are co-location relationships, co-work relationships, mutuality and centrality, and cooperativeness. An unsupervised learning technique was employed to label the data's trustworthiness. After the labeling, an SVM model predicts the trust level. Experiments were performed to observe the performance of the proposed solution. The authors do not specify the threat model. They offer a metric that provides a perception of a node before interaction. This can be used when there have been no previous interactions or a new node has just entered the network.

Bao and Chen [17] consider an IoT environment with no centralized trusted authority. Every device (node) has an owner, and an owner could have many devices. Each owner has a list of friends, representing their social relationships. The trust properties used for trust calculation are honesty, cooperativeness, and community interest. The threat model of this approach includes BMA, SPA, and BSA. The experiments tested the effect of trust parameters on trust evaluation. One drawback of this approach is the limited threat model to only trust-related attacks.

Chen et al. [24] proposed a trust management protocol for Social IoT systems that can form a community of interests. The trust parameters can be dynamically adapted to changes in the environment. The malicious nodes can perform discrimination attacks: BMA, SPA, WHA, and BSA. Each device has an owner. Each owner has a list of friends, representing their social relationships. The trust parameters taken into account for the trust evaluation procedure are honesty, cooperativeness, and community of interest. Experiments were conducted on the NS-3 simulator to test the performance of the proposed protocol. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks.

Wen et al. [76] proposed a method based on [19], a cluster-based scheme, where the nodes form communities of interest. The whole network is governed by a SIOT server. In this work, trust is evaluated from both direct and indirect trust. They introduced a deep learning model to predict the trust value of the new nodes to solve the cold-start problem. The malicious nodes can perform BMA, BSA, and OOA. The following experiments took place: accuracy with a different number of malicious nodes, comparison with another model, and adding a new node to the network. The final results were presented in diagrams. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach deals with the cold-start problem, an important issue for newcomers in a network.

Ben Abderrahim et al. [19] proposed a cluster approach. The nodes form communities of interest. The network is governed by an SIOT server. This approach detects OOA through the use of a Kalman Filter. The malicious nodes can perform BMA, BSA, and OOA. The outcome of the transactions is taken into account to calculate the trust, as are the previous trust values. The code was developed using Python programming language to identify the best weights under which the estimated trust is close to the objective one, observe the performance during OOA, and perform experiments on trust prediction. The results are presented in diagrams. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks.

Aalibagi et al. [1] presented a social network using a bipartite graph. Assuming that there are a finite number of service types, they construct one for every service. The bipartite graph consists of two sets of nodes: trustors U and trustees V. In the bipartite graph, trustor u has an edge linked to trustee v if u has already used the services provided by v at least once. The edge is decorated with a weight (number) indicating the trustor's trust experience while using services provided by the trustee. As a next step, they are finding trust similarities between trustors based on their past experiences, similarity, and centrality measures. To measure the similarity, the Hellinger similarity, the Bayesian similarity, and the connection similarity are used. Also, two other metrics for centrality are used. Afterward, matrix factorization is used to predict the trustworthiness of a trustee. The method takes into account the friend's feedback based on how similar the two nodes are. This method also considers the cold-start problem. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach deals with the cold-start problem, an important issue for newcomers in a network.

Magdich et al. [51] are working on a network of service requestors (SR) and service providers (SP), where the SR evaluates the trust experience with the SP. The trust computation relies on QoS and social metrics. The nodes are also taking into account recommendations from other nodes. After forming the trust score, the nodes act for attack detection. The goal is to identify the attacker and predict the attack it's performing. To achieve it Machine learning and Deep Learning techniques were applied using the following features: reputation, recommendation, similarity, knowledge, and trust. Experiments were performed using the Cooja simulator. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach is tested on real social data and experiments made with different Machine learning and Deep Learning techniques.

Magdich et al. [52] are considering an environment consisting of a set of users and a set of devices. A user can own one or multiple devices, and the devices can provide or request one or more services. Every device is represented by a vector containing three values: user, device, and environment (public or private). The environment value sets the threshold of trust. Also, each device stores its characteristics (manufacturer, type, capacity, and location),

the profile of its owners (friendship, CoI, and Co-work), the transaction history between other nodes, and trust values. In this approach, the trust of the owner, the device, and the environment are taken into account to decide the trust value. The method also uses recommendations. Afterward, the threshold of trust has to be decided. The authors are proposing a Machine Learning technique that they compare with the static method available in the literature. The ML (Artificial Neural Network) algorithm classifies the nodes as trustworthy or not. After each interaction, the nodes evaluate each other and share the result with the other nodes as a recommendation. The authors do not specify the threat model, but they take the context into account.

H. PREDICTION

Sometimes the image of the trustworthiness of a node is not clear, so a prediction mechanism would help the rest of the nodes estimate its trust level. There are works for trust management that use prediction mechanisms to enhance the trust evaluation process. The summary of the findings for this category is presented in Table 13.

We can observe from the Table that 22.2% of the paper uses only direct trust for information gathering, while the rest also uses recommendations. Regarding the Trust Update: 22.2% of the papers use a Time-Driven approach, 33.3% use an Event-Driven approach, and the rest do not refer to the Trust Update. All of the papers present Experiments, but 55.5% of them refer to the simulator used. Also, all of the papers use Multi-Trust for Trust Formation, and none of them use a Central Authority for trust calculation. Finally, 55.5% of the papers define the Threat Model.

Wang et al. [74] proposed a trust model based on direct and indirect trust computation with trust prediction. The prediction method depends on the combination of exponential smoothing and a Markov chain. Exponential smoothing was employed to predict trust and a Markov chain was employed to fix any deviation. Thus, a prediction method was employed to predict the current trust level based on interaction history, behavior history, and some other factors like the device model. For the trust computation, both social and unsocial parameters were considered. The following experiments were performed: comparison of trust prediction with different exponential smoothing coefficients; comparison between first and second exponential smoothing; and experiments for different kinds of attacks. The results were presented as diagrams. The authors do not specify the threat model, which is a drawback. On the other hand, they are proposing a solution to the communication latency issue.

Subhash et al. [72] proposed the Power Trust. Power Trust assigns trust values to the network nodes based on energy auditing. Using the energy auditing model, they calculate the trust values of every node present in the network dynamically and predict physical and cyberattacks. To detect the attacks, a deep learning model was trained with past data that contains normal and excessive energy consumption due to an attack. The model can predict both physical and cyberattacks. The experiments were performed in the Cooja simulator, and they were focused on the performance and the accuracy of the method. The results were presented in diagrams. The authors do not state the threat model, and they do not give details about the deep learning model used. The method takes into account energy consumption, which is important in a resource-constrained environment. Finally, they also predict both physical and cyberattacks.

Wen et al. [76] proposed a method based on [19], a cluster-based scheme, where the nodes form communities of interest. The whole network is governed by an SIOT server. In this work, trust is evaluated from both direct and indirect trust. They introduced a deep learning model to predict the trust value of the new nodes to solve the cold-start problem. The malicious nodes can perform BMA, BSA, and OOA. The following experiments took place: accuracy with a different number of malicious nodes, comparison with another model, and adding a new node to the network. The final results were presented in diagrams. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach deals with the cold-start problem, an important issue for newcomers in a network.

Alnumay et al. [5] proposed a cluster-based system where every cluster has a cluster head. The proposed trust model combines both direct and indirect trust. A Beta probabilistic distribution and the theory of ARMA/GARCH are used to combine the trust units and derive the trust value. The cluster heads can predict the trust value ahead using this method. A malicious node can perform the following attacks: SFA, Routing table overflow and resource consumption attacks, Byzantine, BA, DoS attacks, and SDA. The trust is computed based on the number of packets properly forwarded, the number of packets dropped, and the number of packets falsely injected. Experiments were performed to observe the performance and accuracy of the proposed solution. Also, they presented some diagrams depicting the comparison with two other models. The main limitation of the approach is that during the initialization of the clusters, a malicious node might become cluster head at first. On the other side, the authors are dealing with a high-mobility environment and can predict multiple steps ahead.

Abderrahim et al. [19] proposed a cluster approach. The nodes form communities of interest. The network is governed by a SIOT server. This approach detects OOA through the use of a Kalman Filter. The malicious nodes can perform BMA, BSA, and OOA. The outcome of the transactions is taken into account to calculate the trust, as are the previous trust values. The code was developed using Python programming language to identify the best weights under which the estimated trust is close to the objective one, observe the performance during OOA, and perform experiments on trust prediction. The results are presented in diagrams. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks.

Jayasinghe et al. [42] proposed a system where the nodes form communities of interest. In this model, the transactions

TABLE 12. Overview of approaches proposing social based trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[65]	Both	Event-Driven	✓		Multi-Trust	✓		• Only trust-related attacks	• Proposal for trust storage • Takes into account the context
[59]	Both	Event-Driven	✓		Multi-Trust			• Pretrusted entities are involved • Do not provide the threat model	• Provides a view of the trust of the whole system
[10]	Both	Time-Driven	✓	✓	Multi-Trust			• Bootstrapping issue with forming clusters the trust values are required • Do not provide the threat model	• Takes into account the data trust
[54]	Both	Event-Driven	✓		Multi-Trust	✓		• Deals only with trust-related attacks	• At steady-state only one parameter is used to compute trust
[2]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account the context
[58]	Both	Event-Driven	✓		Multi-Trust			• Do not specify the threat model	
[74]	Both		✓		Multi-Trust			• Do not specify the threat model	• Dealing with the communication latency
[25]	Both	Event-Driven	✓		Single-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Section for trust storage
[16]	Both	Event-Driven	✓		Multi-Trust	✓		• Deals only with a set of trust-related attacks	• Tests scalability for storing the trust values
[47]	Both	Event-Driven	✓		Multi-Trust			• Do not specify the threat model	• Ensure secure communication using secret codes
[3]	Both	Event-Driven	✓		Multi-Trust			• Do not specify the threat model	• Takes into account the context
[18]	Both	Event-Driven	✓			✓		• Deals only with a set of trust-related attacks	• Propose a storage management strategy
[26]	Both		✓		Multi-Trust		✓	• Do not specify the threat model	• Tests the energy consumption
[11]	Both	Both		✓	Multi-Trust			• Purely theoretical model • Do not specify the threat model	• Cross-domain trust management • Extensive literature review
[9]	Both	Event-Driven	✓		Single-Trust	✓		• Do not study blockchain-related attacks	• Dynamically selected counselors • Cost analysis in terms of gas
[42]	Direct		✓		Multi-Trust			• Do not specify the threat model	• Metric that provides a perception of a node before interaction
[17]	Both	Event-Driven	✓		Multi-Trust	✓		• Deals only with a set of trust-related attacks	
[24]	Both	Event-Driven	✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Dynamically adjustable
[76]	Both	Event-Driven	✓		Multi-Trust	✓		• Deals only with a set of trust-related attacks	• Deals with cold-start problem
[19]	Both	Time-Driven	✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	•
[1]	Both		✓		Multi-Trust	✓		• Deals only with a set of trust-related attacks	• Deals with cold-start problem
[51]	Both	Time-Driven	✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Experiments with real social data • Different Models were tested
[52]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account the context

are under evaluation and should be determined if a transaction is trustworthy. The trust parameters used are co-location relationships, co-work relationships, mutuality and centrality, and cooperativeness. An unsupervised learning technique was employed to label the data's trustworthiness. After the labeling, an SVM model predicts the trust level. Experiments were performed to observe the performance of the proposed solution. The authors do not specify the threat model. They offer a metric that provides a perception of a node before interaction. This can be used when there have been no previous interactions or a new node has just entered the network.

Aalibagi et al. [1] presented a social network using a bipartite graph. Assuming that there are a finite number of service types, they construct one for every service. The bipartite graph consists of two sets of nodes: trustors U and trustees V. In the bipartite graph, trustor u has an edge linked to trustee v if u has already used the services provided by v at least once. The edge is decorated with a weight (number) indicating the trustor's trust experience while using services provided by the trustee. As a next step, they are finding trust similarities between trustors based on their past experiences, similarity, and centrality measures. To measure the similarity, the Hellinger similarity, the Bayesian similarity, and the connection similarity are used. Also, two other metrics for centrality are used. Afterward, matrix factorization is used to predict the trustworthiness of a trustee. The method takes into account the friend's feedback based on how similar the two nodes are. This method also considers the cold-start problem. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach deals with the cold-start problem, an important issue for newcomers in a network.

Magdich et al. [51] are working on a network of service requestors (SR) and service providers (SP), where the SR evaluates the trust experience with the SP. The trust computation relies on QoS and social metrics. The nodes are also taking into account recommendations from other nodes. After forming the trust score, the nodes act for attack detection. The goal is to identify the attacker and predict the attack it's performing. To achieve it Machine learning and Deep Learning techniques were applied using the following features: reputation, recommendation, similarity, knowledge, and trust. Experiments were performed using the Cooja simulator. The main limitation of the approach is that the threat model is limited to a set of trust-related attacks. On the other side, the approach is tested on real social data and experiments made with different Machine learning and Deep Learning techniques.

Magdich et al. [52] are considering an environment consisting of a set of users and a set of devices. A user can own one or multiple devices, and the devices can provide or request one or more services. Every device is represented by a vector containing three values: user, device, and environment (public or private). The environment value sets the threshold of trust. Also, each device stores its characteristics (manufacturer, type, capacity, and location), the profile of its owners (friendship, CoI, and Co-work), the transaction history between other nodes, and trust values. In this approach, the trust of the owner, the device, and the environment are taken into account to decide the trust value. The method also uses recommendations. Afterward,

the threshold of trust has to be decided. The authors are proposing a Machine Learning technique that they compare with the static method available in the literature. The ML (Artificial Neural Network) algorithm classifies the nodes as trustworthy or not. After each interaction, the nodes evaluate each other and share the result with the other nodes as a recommendation. The authors do not specify the threat model, but they take the context and social aspects into account.

I. PROBABILISTIC APPROACHES

There has been some research on estimating trust based on probability theory. A summary of the findings for this category is summarized in Table 14.

We can observe from the Table that 33.3% of the papers are using only Direct observations, while the rest of them (66.7%) are also using recommendations for information gathering. Also, 16.7% of the papers preferred a Time-Driven approach, 16.7% an Event-driven one, and the rest of them did not refer to the Trust Update. All of the papers present Experiments, but 66.6% of them refer to the Simulator used. Also, none of the papers is using a Centralized entity for trust evaluation. Regarding Trust Formation, 66.6% are using Multi-Trust, while the rest do not refer to the Trust Update. Finally, 50% of them defined the Threat Model.

Boudagdigue et al. [21] proposed a system where every node is monitored by its neighbors. Also, some groups of neighbors are formed to evaluate indirect trust. This paper proposes a distributed trust model based on a similar model proposed for vehicular networks. The authors used Markov Chains to model the trust change. A discrete-time chain with $M + 1$ states was introduced. State 0 corresponds to the lower trust level and M to the uppermost. The probabilities were calculated based on the direct and indirect trust scores. The malicious nodes can perform BMA, BSA, selfish attacks, and honesty attacks. The trust parameters taken into account are honesty and cooperation. The experiments tested the proposed solution against different kinds of trust-related attacks. The results were presented as diagrams. One limitation of the current solution is that it only deals with a set of trust-related attacks. Also, the authors do not provide details about the experimental environment. On the other hand, the authors are improving an existing method proposed for VANETS in the IoT context.

Wang et al. [74] proposed a trust model based on direct and indirect trust computation with trust prediction. The prediction method depends on the combination of exponential smoothing and a Markov chain. Exponential smoothing was employed to predict trust and a Markov chain was employed to fix any deviation. Thus, a prediction method was employed to predict the current trust level based on interaction history, behavior history, and some other factors like the device model. For the trust computation, both social and unsocial parameters were considered. The following experiments were performed: comparison of trust prediction with different exponential smoothing coefficients; comparison between first and second exponential smoothing; and experiments for different kinds of attacks. The results were presented as diagrams. The authors do not specify the threat model, which is a drawback. On the other hand, they are proposing a solution to the communication latency issue.

Joshi et al. [44] proposed a system that consists of several resource-constrained IoT nodes with a short radio range and a base station with a limitless source of energy as a central authority. This research work has presented a 2-state HMM with a Trusted state and a compromised state, together with essential and unessential output as observation states. The trustworthiness of the node is modeled by the 2-state HMM to predict the likelihood of the node's next state. The state transition probability matrix is defined by the energy consumed, the number of modified packets, and the number of forwarding packets. The malicious nodes can drop the packets or tamper with the data. Experiments were conducted in MATLAB to evaluate the network's trustworthiness with various percentages of compromised nodes and compare it with other methods. The results were presented in diagrams. The authors are taking into account only two kinds of attacks. The authors are using energy consumption as a key characteristic for calculating trust. This is interesting since increased activity might be malicious, but also energy of the nodes is taken into account in a resource-constrained environment.

Wang et al. [75] proposed a system consisting of Mobile edge nodes (MEN) and common sensors. The MEN are connected to a small number of sensors. In this paper, a mobile edge trust evaluation scheme is proposed. The evaluation of the trustworthiness of sensor nodes is achieved using a probabilistic graph model. The probabilistic graph model is used to represent the relationship between nodes. The interaction of node i with node j can be described as P and Q. P is a positive influence of node i on node j and Q is a negative one. The information gathered for the formation of trust is the result of data collection and communication behavior. Also, a moving strategy method is proposed to decrease the travel distance MEN has to cover to evaluate every sensor. The experiments were conducted using MATLAB and NS-3 and were focused on the performance of the mechanism, the analysis of energy consumption, and the testing of the proposed moving algorithm. The results were presented in diagrams. The paper does not specify the threat model, which is a drawback. A strength, on the other hand, is that the authors propose a moving strategy for energy savings in a high-mobility environment.

Fang et al. [35] proposed a system with a cluster-based architecture. The paper proposes a trust management scheme using Dirichlet Distribution. Both direct observations and third-party recommendations are considered to calculate the trust value of a node. This work is proposed to defend against internal attacks. Experiments conducted on MATLAB focus on the comparison with Beta distribution-based and Gaussian distribution-based performance experiments for OOA. The final results were presented in diagrams. One limitation of

TABLE 13. Overview of approaches proposing prediction based trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[74]	Both		✓		Multi-Trust			• Do not specify the threat model	• Dealing with the communication latency
[72]	Direct		✓		Multi-Trust		✓	• Do not specify the threat model • Do not give details about the deep learning	• Predict the physical attacks and cyber-attacks • Focused on energy consumption
[76]	Both	Event-Driven	✓		Multi-Trust	✓		• Deals only with a set of trust-related attacks	• Deals with cold-start problem • Takes into account social aspects
[5]	Both	Event-Driven	✓		Multi-Trust	✓		•At initialization a malicious node can be CH	• Predicts multi-step ahead • Mobility
[19]	Both	Time-Driven	✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Takes into account social aspects
[42]	Direct		✓		Multi-Trust			• Do not specify the threat model	• Metric that provides a perception of a node before interaction
[1]	Both		✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Deals with cold-start problem
[51]	Both	Time-Driven	✓		Multi-Trust	✓	✓	• Deals only with a set of trust-related attacks	• Experiments with real social data • Different Models were tested
[52]	Both	Event-Driven	✓		Multi-Trust		✓	• Do not specify the threat model	• Takes into account social aspects and the context

TABLE 14. Overview of approaches proposing probabilistic and markov chain-based trust management methods.

Paper	Info Gathering	Trust Update	Experiments	Centralized	Trust Formation	Threat Model	Simulator	Limitations	Strengths
[21]	Both		✓		Multi-Trust	✓		• Doesn't provide details for the experimental environment • Deals with a set of trust-related attacks	• Improving an existing method
[74]	Both		✓		Multi-Trust			• Do not specify the threat model	• Dealing with the communication latency
[44]	Direct	Time-Driven	✓		Multi-Trust	✓	✓	• Deals with only 2 kind of attacks	• Takes into account the energy consumption
[75]	Direct		✓		Multi-Trust		✓	• Do not specify the threat model	• Travel strategy for energy saving • Mobility
[35]	Both		✓				✓	• Do not specify the threat model • Doesn't specify when a transaction is a success, fail or uncertain.	• Gives a solution for the cluster head selection
[34]	Both	Event-Driven	✓			✓	✓	• Only one attack is considered	• Cyber-security requirements for ICN

the solution is that the authors do not specify the threat model. Also, they do not specify when a transaction is successful or not. On the other hand, they propose a solution for cluster head selection.

Fang et al. [34] proposed a trust management technology that guards the system against OOA. Also, Beta distribution is used for the trust evaluation procedure. The authors also mentioned the cyber-security requirements for Information-Centric Networking (ICN). Experiments were conducted on MATLAB to observe the performance of the scheme and compare it with other techniques. One drawback of the method is that it only covers one attack. On the other hand, the authors are studying the cyber-security requirements for ICN systems.

VII. CHALLENGES

Based on the above analysis, we are presenting some highlights of the vulnerabilities we observed.

- *Scalability:* We can observe from the above analysis that most of the works did not take into account the scalability factor when conducting the experiments. In extended IoT networks, where a huge number of sensors are connected and communicating with each other, a trust management system should be able to respond efficiently. Especially in dynamic networks, where the number of nodes is not fixed, a trust management scheme should be able to adapt to a growing amount of work.
- *Privacy:* Privacy is a really important factor in every system. Especially for resource-constrained IoT devices. A trust management system may handle sensitive data to calculate and preserve the trust between two nodes. For example, the frequency of communication between the nodes. Also, blockchain technology can solve multiple problems, but a public blockchain that offers decentralization lacks privacy preservation. If some sensitive piece of information has to be exposed in a smart contract or stored on a blockchain, it is visible to all the participants.
- *Context:* Context is really important for trust. Some individuals are to be trusted in specific contexts or circumstances. A malicious node may be trustworthy only in a specific context. Only five papers included in this survey take context into account to calculate trust. The highly heterogeneous IoT networks act differently in different contexts.
- *Energy:* Energy is a really important factor for resource-constrained IoT devices. The research community hasn't extensively investigated the issue of designing a lightweight trust management system. We observed that a few of the works are conducting experiments to measure the energy consumption caused by the operation of trust management.
- *Attacks:* The trust management systems add some trust-related attacks to the threat model. These kinds of attacks should be tackled by the trust management system to be valuable for the IoT network. During the analysis, we saw that most of the works are mainly dealing with trust-related attacks. The attacks that gained the most attention from the research community are BMA, BSA, and SPA. The OOA and OSA have not been studied so extensively. However, a trust management scheme should be able to detect multiple attacks, not just

trust-oriented ones. Especially, blockchain-based works that refer to the IoT nodes as lightweight nodes should take the EA into account. In future work, it will be nice to investigate trust management techniques that tackle trust-related attacks but also deal with a variety of other attacks.

- *Mobility:* IoT can also be involved in high-mobility tasks (e.g. smart vehicles). Designing a trust-management system that can be adjusted to a high-mobility environment is important. There is not a lot of work involved in dealing with this issue.
- *Cold-start problem:* The initialization of the trust values during bootstrapping is an issue that has to be addressed. This problem also occurs when a new node enters the network. This is an interesting and important issue that has to be addressed, especially in dynamic environments where nodes come and go constantly.
- *Threat Model:* We have noticed that there are a lot of works that do not refer to the threat model. In our opinion, it is highly important to state the threat model and the attacks the proposed trust management technique is tackling. Different kinds of attacks are suitable for each system.
- *Pre-trusted entities:* some approaches require some pre-trusted entities. This is an assumption that might not be applicable in real-life scenarios. In case of such an assumption, it should be clearly stated which are the pre-trusted entities and their role in the system. However, these approaches are not always applicable.
- *Detailed analysis of technologies used:* It is important to give detailed information about the components used in the system. Especially in the case where other technologies are used, for example, blockchain and machine learning. The characteristics of the blockchain should be given in detail since a minor assumption on the reader's side can change the whole system.
- *Defence Mechanisms:* If a malicious node is detected, there should be some defense mechanisms. Some works propose the exclusion of the node of the network. The research community should focus more on this subject to propose better alternatives.
- *Filtering Recommendations:* Filtering recommendations are important in cases where indirect trust is enabled. Filtering recommendations can help prevent trust-related attacks.
- *Edge and Cloud Architectures:* Nowadays, IoT is connected with Edge and Cloud computing. These architectures can be exploited to design trust management methods and take some computational and storage weight of the resource-constrained IoT devices.
- *Data Trust:* Some works are taking into account the data trust. This concept can help design trust management techniques that take into account general attacks (not only trust-related). Data tampering can cause major issues with the functionality of an application (e.g. e-health).
- *Storage:* Another important issue is storage. In some cases, nodes have to hold huge amounts of trust-related information. There are some works dealing with this issue, proposing some efficient storage mechanisms, but the field needs further investigation.
- *Network Traffic:* There are several techniques aimed at detecting malicious traffic in IoT networks (see [69] and the references therein). We believe that those techniques could constitute the basis of new trust management methods that could use the traffic classification parameters as the basis of the trust and reputation metrics.

VIII. CONCLUSION

Security and trust are critical in IoT systems since devices often run in potentially hostile environments. One way to assess the trustworthiness of a node is by using trust management techniques. Indeed, trust management in IoT has gained a lot of researchers' attention in the last decade resulting in a vast literature that is not easy to navigate. Motivated by this challenge, our work has addressed several research questions:

- RQ1 - Which methods are currently used in the field? Our work answers this question by providing a structured overview of a comprehensive set of the literature in the field. Our survey follows a well-disciplined approach to systematic literature reviews, which would allow any interested researcher to update and reproduce our main findings. The structure of our overview follows a classification based on categories and dimensions that have emerged from our preliminary analysis. Table 5 provides a bird's eye view of all the papers covered, which are summarized and discussed in Section VI. The main outcome is not just a guide to existing works but it also highlights which classes of approaches are more predominant (e.g. social-based approaches using direct and indirect information-gathering techniques) and which ones are less explored (e.g. centralized approaches).
- RQ2 - What is the threat model of those proposals? We answer this question by considering a wide class of attacks identified in the literature on IoT security and how the proposed approaches covered in the survey relate to them (see Section VI). Our main conclusion is that most proposals are not specific in the threat model being considered and that most of the works focus on data integrity and trust-related attacks. In general, the are lot of potential attacks are not covered extensively by the intended threat models.
- RQ3 - What are the strengths and limitations of each proposal? Our work answers this question in VI by providing the strengths and limitations of each specific paper covered by the survey. It emerges from our study, for example, that many approaches proposed in the

literature do not specify the intended threat model, which makes it difficult to understand the assumptions under which the proposed approach will be effective.

- RQ4 - What are the challenges and future directions? We answer this question in Section VII, where discuss a list of areas such as privacy, scalability, and bootstrapping, and the challenges related to trust management in IoT. One example of challenge and future direction regards the lack of use of precise threat models that emerged from our evaluation of the limitations of existing approaches. We believe that the discussion in Section VII can help researchers find interesting areas in need of investigation and development.

Overall, we believe that our work can help readers decide on the methods and technologies that are more suitable for a particular IoT trust management mechanism and the challenges that should be considered when designing such a system. Our work can also support researchers in identifying future research avenues. An example would be the use of attack-defense trees to build solid trust management systems, containing all possible attacks and countermeasures.

REFERENCES

[1] 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 Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2274–2285, Jul. 2022.

[2] R. Abidi and N. B. Azzouna, "Self-adaptive trust management model for social IoT services," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2021, pp. 1–7.

[3] A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, Á. MacDermott, and X. Wang, "CTRUST: A dynamic trust model for collaborative applications in the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5432–5445, Jun. 2019.

[4] H. Aldawsari and A. M. Artoli, "A reliable lightweight trust evaluation scheme for IoT security," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, pp. 723–731, 2021.

[5] W. Alnumay, U. Ghosh, and P. Chatterjee, "A trust-based predictive model for mobile ad hoc network in Internet of Things," *Sensors*, vol. 19, no. 6, p. 1467, Mar. 2019.

[6] M. D. Alshehri and F. K. Hussain, "A comparative analysis of scalable and context-aware trust management approaches for Internet of Things," in *Neural Information Processing*, S. Arik, T. Huang, W. K. Lai, Q. Liu, Eds. Cham, Switzerland: Springer, 2015, pp. 596–605.

[7] M. D. Alshehri and F. K. Hussain, "A fuzzy security protocol for trust management in the Internet of Things (fuzzy-IoT)," *Computing*, vol. 101, no. 7, pp. 791–818, Jul. 2019.

[8] A. Altaf, H. Abbas, F. Iqbal, F. A. Khan, S. Rubab, and A. Derhab, "Context-oriented trust computation model for industrial Internet of Things," *Comput. Electr. Eng.*, vol. 92, Jun. 2021, Art. no. 107123.

[9] M. Amiri-Zarandi, R. Dara, and E. Fraser, "LBTM: A lightweight blockchain-based trust management system for social Internet of Things," *J. Supercomput.*, vol. 78, pp. 1–19, Apr. 2022.

[10] S. F. A. Mon, S. G. Winster, and R. Ramesh, "Trust model for IoT using cluster analysis: A centralized approach," *Wireless Pers. Commun.*, vol. 127, no. 1, pp. 715–736, Nov. 2022.

[11] K. A. Awan, I. U. Din, M. Zareei, M. Talha, M. Guizani, and S. U. Jadoon, "HoliTrust—A holistic cross-domain trust management mechanism for service-centric Internet of Things," *IEEE Access*, vol. 7, pp. 52191–52201, 2019.

[12] K. A. Awan, I. U. Din, A. Almogren, M. Guizani, A. Altameem, and S. U. Jadoon, "RobustTrust—A pro-privacy robust distributed trust management mechanism for Internet of Things," *IEEE Access*, vol. 7, pp. 62095–62106, 2019.

[13] N. Bansal, "IoT applications in agriculture," Apress, Berkeley, CA, USA, Tech. Rep., 2020, pp. 93–114.

[14] N. Bansal, "IoT applications in energy," Apress, Berkeley, CA, USA, Tech. Rep., 2020, pp. 115–134.

[15] N. Bansal, "IoT applications in transportation," Apress, Berkeley, CA, USA, Tech. Rep., 2020, pp. 239–262.

[16] F. Bao and I. R. Chen, "Dynamic trust management for Internet of Things applications," in *Proc. Int. Workshop Self-Aware Internet Things (Self-IoT)*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 1–6.

[17] F. Bao and I.-R. Chen, "Trust management for the Internet of Things and its application to service composition," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2012, pp. 1–6.

[18] F. Bao, I.-R. Chen, and J. Guo, "Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems," in *Proc. IEEE 11th Int. Symp. Auton. Decentralized Syst. (ISADS)*, Mar. 2013, pp. 1–7.

[19] 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 *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 747–752.

[20] B. Bordel and R. Alcarria, "Distributed trust and reputation services in pervasive Internet-of-Things deployments," in *Mobile Internet Security*, I. You, H. Kim, T. Y. Youn, F. Palmieri, I. Kotenko, Eds. Singapore: Springer, 2022, pp. 16–29.

[21] C. Boudagdigue, A. Benslimane, A. Kobbane, and M. Elmachkour, "A distributed advanced analytical trust model for IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[22] O. Castillo and P. Melin, *Fuzzy Logic*. Heidelberg, Germany: Physica-Verlag, 2001, pp. 5–27.

[23] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for Internet of Things," *Comput. Sci. Inf. Syst.*, vol. 8, no. 4, pp. 1207–1228, Oct. 2011.

[24] I.-R. Chen, F. Bao, and J. Guo, "Trust-based service management for social Internet of Things systems," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 6, pp. 684–696, Nov. 2016.

[25] I.-R. Chen, J. Guo, and F. Bao, "Trust management for SOA-based IoT and its application to service composition," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 482–495, May 2016.

[26] R. Das, M. Singh, and K. Majumder, "SGSQoT: A community-based trust management scheme in Internet of Things," in *Proc. eHaCON*, Kolkata, India: Springer, Jan. 2019, pp. 209–222.

[27] M. Debe, K. Salah, R. Jayaraman, I. Yaqoob, and J. Arshad, "Trustworthy blockchain gateways for resource-constrained clients and IoT devices," *IEEE Access*, vol. 9, pp. 132875–132887, 2021.

[28] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, "A trust architecture for blockchain in IoT," in *Proc. 16th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services (MobiQuitous)*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 190–199.

[29] I. U. Din, K. A. Awan, A. Almogren, and B.-S. Kim, "ShareTrust: Centralized trust management mechanism for trustworthy resource sharing in industrial Internet of Things," *Comput. Electr. Eng.*, vol. 100, May 2022, Art. no. 108013.

[30] 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 Things J.*, vol. 10, no. 4, pp. 2776–2783, Feb. 2023.

[31] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for IoT security," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102467.

[32] J. Duan, D. Gao, D. Yang, C. H. Foh, and H.-H. Chen, "An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 58–69, Feb. 2014.

[33] C. Esposito, O. Tamburis, X. Su, and C. Choi, "Robust decentralised trust management for the Internet of Things by using game theory," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102308.

[34] W. Fang, M. Xu, C. Zhu, W. Han, W. Zhang, and J. J. P. C. Rodrigues, "FETMS: Fast and efficient trust management scheme for information-centric networking in Internet of Things," *IEEE Access*, vol. 7, pp. 13476–13485, 2019.

[35] W. Fang, W. Zhang, L. Shan, X. Ji, and G. Jia, "DDTMS: Dirichlet-distribution-based trust management scheme in Internet of Things," *Electronics*, vol. 8, no. 7, p. 744, Jul. 2019.

[36] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[37] S. Guleng, C. Wu, X. Chen, X. Wang, T. Yoshinaga, and Y. Ji, "Decentralized trust evaluation in vehicular Internet of Things," *IEEE Access*, vol. 7, pp. 15980–15988, 2019.

[38] J. Guo, I.-R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in Internet of Things systems," *Comput. Commun.*, vol. 97, pp. 1–14, Jan. 2017.

[39] M. Hürlimann, *Fuzzy Logic*. Wiesbaden, Germany: Gabler, 2009, pp. 41–58.

[40] M. J. Jabraeil, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, "IoT architecture," in *Towards the InternetofThings:Architectures, Security, and Applications*. Cham, Switzerland: Springer, Jan. 2020, pp. 9–31.

[41] L. S. Jayashree and G. Selvakumar, *Edge Computing in IoT*. Cham, Switzerland: Springer, 2020, pp. 49–69.

[42] U. Jayasinghe, G. M. Lee, T.-W. Um, and Q. Shi, "Machine learning based trust computational model for IoT services," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 39–52, Jan. 2019.

[43] F. Jeribi, R. Amin, M. Alhameed, and A. Tahir, "An efficient trust management technique using ID3 algorithm with blockchain in smart buildings IoT," *IEEE Access*, vol. 11, pp. 8136–8149, 2023.

[44] G. Joshi and V. Sharma, "Light-weight hidden Markov trust evaluation model for IoT network," in *Proc. 5th Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Nov. 2021, pp. 142–149.

[45] B. A. Kitchenham and S. Charter, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ., Durham Univ. Joint Report, Tech. Rep. EBSE 2007-001, Jul. 2007.

[46] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, "Decentralized blockchain-based trust management protocol for the Internet of Things," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1292–1306, Mar. 2022.

[47] A. M. Kowshalya and M. L. Valarmathi, "Dynamic trust management for secure communications in social Internet of Things (SIoT)," *Sādhanā*, vol. 43, no. 9, p. 136, Sep. 2018.

[48] R. Kumar and R. Sharma, "Leveraging blockchain for ensuring trust in IoT: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8599–8622, Nov. 2022.

[49] Q. Lin and Q. Zhao, "IoT applications in healthcare," in *Internet of Things*. Cham, Switzerland: Springer, 2021, pp. 115–133.

[50] Z. Ma, L. Liu, and W. Meng, "DCONST: Detection of multiple-mix-attack malicious nodes using consensus-based trust in IoT networks," in *Information Security and Privacy*, J. K. Liu and H. Cui, Eds. Cham, Switzerland: Springer, 2020, pp. 247–267.

[51] R. Magdich, H. Jemal, and M. B. Ayed, "A resilient trust management framework towards trust related attacks in the social Internet of Things," *Comput. Commun.*, vol. 191, pp. 92–107, Jul. 2022.

[52] R. Magdich, H. Jemal, and M. Ben Ayed, "Context-awareness trust management model for trustworthy communications in the social Internet of Things," *Neural Comput. Appl.*, vol. 34, no. 24, pp. 21961–21986, Dec. 2022.

[53] M. Mahmud, M. S. Kaiser, M. M. Rahman, M. A. Rahman, A. Shabut, S. Al-Mamun, and A. Hussain, "A brain-inspired trust management model to assure security in a cloud based IoT framework for neuroscience applications," *Cognit. Comput.*, vol. 10, no. 5, pp. 864–873, Oct. 2018.

[54] C. Marche and M. Nitti, "Trust-related attacks and their detection: A trust management model for the social IoT," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 3297–3308, Sep. 2021.

[55] C. V. L. Mendoza and J. H. Kleinschmidt, "A distributed trust management mechanism for the Internet of Things using a multi-service approach," *Wireless Pers. Commun.*, vol. 103, no. 3, pp. 2501–2513, Dec. 2018.

[56] M. S. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions," *Comput. Commun.*, vol. 139, pp. 32–57, May 2019.

[57] S. Nakamoto. (Mar. 2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://metzdowd.com

[58] M. Nitti, R. Girau, L. Atzori, A. Iera, and G. Morabito, "A subjective model for trustworthiness evaluation in the social Internet of Things," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2012, pp. 18–23.

[59] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social Internet of Things," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1253–1266, May 2014.

[60] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.

[61] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, and R. Brooks, "The sleep deprivation attack in sensor networks: Analysis and methods of defense," *Int. J. Distrib. Sensor Netw.*, vol. 2, no. 3, pp. 267–287, Jul. 2006.

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

[63] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-based blockchain authorization for IoT," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1646–1658, Jun. 2021.

[64] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, "Trust management and evaluation for edge intelligence in the Internet of Things," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103756.

[65] S. E. A. Rafey, A. Abdel-Hamid, and M. A. El-Nasr, "CBSTM-IoT: Context-based social trust model for the Internet of Things," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, Apr. 2016, pp. 1–8.

[66] R. Rani, S. Kumar, and U. Dohare, "Trust evaluation for light weight security in sensor enabled Internet of Things: Game theory oriented approach," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8421–8432, Oct. 2019.

[67] M. Saeed, M. Aftab, R. Amin, and D. Koundal, "Trust management model in IoT: A comprehensive survey," in *Innovations in Bio-Inspired Computing and Applications*, A. Abraham, A. M. Madureira, A. Kaklauskas, N. Gandhi, A. Bajaj, A. K. Muda, D. Kriksciuniene, J. C. Ferreira, Eds. Cham, Switzerland: Springer, 2022, pp. 675–684.

[68] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach," *Comput. Secur.*, vol. 39, pp. 351–365, Nov. 2013.

[69] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and bot-IoT attacks traffic identification for Internet of Things in smart city," *Future Gener. Comput. Syst.*, vol. 107, pp. 433–442, Jun. 2020.

[70] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, "Towards trustworthy Internet of Things: A survey on trust management applications and schemes," *Comput. Commun.*, vol. 160, pp. 475–493, Jul. 2020.

[71] S. Singh and M. Kandpal, "A comprehensive survey on trust management in fog computing," in *ICT Analysis and Applications*, S. Fong, N. Dey, and A. Joshi, Eds. Singapore: Springer, 2022, pp. 87–97.

[72] P. Subhash, G. R. Chandra, and K. S. Surya, "Power trust: Energy auditing aware trust-based system to detect security attacks in IoT," in *Proc. 6th Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2021, pp. 265–269.

[73] H. Tyagi and R. Kumar, *Cloud Computing for IoT*. Cham, Switzerland: Springer, 2020, pp. 25–41.

[74] E. Wang, C. M. Chen, D. Zhao, W. H. Ip, and K. Yung, "A dynamic trust model in Internet of Things," *Soft Comput.*, vol. 24, pp. 5773–5782, Apr. 2020.

[75] T. Wang, H. Luo, W. Jia, A. Liu, and M. Xie, "MTES: An intelligent trust evaluation scheme in sensor-cloud-enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2054–2062, Mar. 2020.

[76] Y. Wen, Z. Xu, R. Zhi, and J. Chen, "Trust prediction model based on deep learning in social Internet of Things," in *IoT as a Service*, B. Li, C. Li, M. Yang, Z. Yan, and J. Zheng, Eds. Cham, Switzerland: Springer, 2021, pp. 557–570.

[77] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*. New York, NY, USA: Association for Computing Machinery, 2014, pp. 1–10.

[78] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.

[79] Y. Yu, Z. Jia, W. Tao, B. Xue, and C. Lee, "An efficient trust evaluation scheme for node behavior detection in the Internet of Things," *Wireless Pers. Commun.*, vol. 93, no. 2, pp. 571–587, Mar. 2017.

ALYZIA MARIA KONSTA received the integrated M.S. degree in electrical and computer engineering from the National Technical University of Athens, in 2020. She is currently pursuing the Ph.D. degree in computer science with the Technical University of Denmark. Her research interests include formal methods, security, and POMDPs.

ALBERTO LLUCH LAFUENTE received the Ph.D. degree from Albert-Ludwigs-Universität Freiburg, in 2003. He is currently a Full Professor with the Technical University of Denmark, where he leads the section for software systems engineering. Previously, he was an Associate Professor with the Technical University of Denmark, an Assistant Professor with the IMT School for Advanced Studies Lucca, and a Postdoctoral Researcher with the University of Pisa. He has participated in several research projects, including CyberSec4Europe (one of the pilots of the European Cybersecurity Competence Centre and Network) where he led activities in secure software lifecycle and cybersecurity education. He has coauthored more than 100 peer-reviewed scientific papers in international journals and conference proceedings. He has edited 17 journal special issues and book chapters. His research interests include formal methods, software engineering, artificial intelligence, and security.

NICOLA DRAGONI received the M.Sc. (cum laude) and Ph.D. degrees in computer science from the University of Bologna, Italy. He is currently a Professor in secure pervasive computing with the DTU Compute, Technical University of Denmark, where he is also the Deputy Director of Research, the Head of the Section (Cybersecurity Engineering), and the Head of the DTU Center for Digital Security (DIGISEC). He has been active in several national and international projects. He has edited three journal special issues and one book. He has coauthored more than 145 peer-reviewed scientific papers in international journals and conference proceedings. His main research interests include pervasive computing and security, with the latest focus on the Internet of Things and fog/edge computing.