Machine Learning Based Trust Computational Model for IoT Services

Upul Jayasinghe, *Student Member, IEEE*,
Gyu Myoung Lee, *Senior Member, IEEE*, Tai-Won Um, and Qi Shi

Abstract—The Internet of Things has facilitated access to a large volume of sensitive information on each participating object in an ecosystem. This imposes many threats ranging from the risks of data management to the potential discrimination enabled by data analytics over delicate information such as locations, interests, and activities. To address these issues, the concept of trust is introduced as an important role in supporting both humans and services to overcome the perception of uncertainty and risks before making any decisions. However, establishing trust in a cyber world is a challenging task due to the volume of diversified influential factors from cyber-physical-systems. Hence, it is essential to have an intelligent trust computation model that is capable of generating accurate and intuitive trust values for prospective actors. Therefore, in this paper, a quantifiable trust assessment model is proposed. Built on this model, individual trust attributes are then calculated numerically. Moreover, a novel algorithm based on machine learning principles is devised to classify the extracted trust features and combine them to produce a final trust value to be used for decision making. Finally, our model's effectiveness is verified through a simulation. The results show that our method has advantages over other aggregation methods.

Index Terms—Clustering, computational trust, feature extraction, knowledge acquisition, model classification

1 Introduction

The concept of the Internet of Things (IoT), which has made many unthinkable inventions possible, has been a major breakthrough in the past decade and many more years to come. In an IoT infrastructure, billions of electronic devices are connected to the Internet and these devices are equipped with sensors that observe or monitor various aspects of human life in the real world for supporting more ubiquitous and intelligent services. A modern day IoT ecosystem involves the networking among physical devices and cyber components as well as the social interactions of them. This is essentially a leap forward of Cyber-Physical Systems (CPS) and the formation of Cyber-Physical-Social Systems (CPSS) to connect the Cyber-Physical world with social world objects [1]. Based on the CPSS concept, the new IoT model, which incorporates social paradigms into the IoT ecosystem, is introduced to explain the social behavior of objects along with human interactions [2].

However, this integration introduces new concerns for risks, privacy and security at both system and social levels as a result of heterogeneous interactions among humans and objects. Consequently, managing risks and securing IoT are broader in scope and pose greater challenges than the traditional privacy and security triad of integrity, confidentiality, and availability in the physical and cyber world. The aim of future IoT services is to make decisions autonomously without human intervention. In this regard, trust has been recognized as a vital key for processing and handling data, and for complying with the services, business, and customer needs. Accordingly, ITU-T has been developing related standards for trust provisioning after publishing the first recommendation [3] based on the activities of Correspondence Group on Trust. For supporting trust, it is crucial to minimize unexpected risks and maximize risk predictability using a trust platform. This platform should help the IoT infrastructure to operate in a controlled manner and to avoid unpredicted conditions and service failures.

Many trust evaluating schemes have been proposed in the literature, beginning from early research work done by Marsh in his dissertation [4]. However, they lack the information about generic framework details, which defines all aspects of trust including information gathering, processing and producing measurable values as the outcome of the platform. Moreover, labeling a particular entity as trustworthy or not based on a given data set of several hundreds of interactions is a vital matter when it comes to feasible deployment. To this end, we have found no such research that has investigated labeling based on trustworthiness. To rectify such a weakness, this paper extends our previous related work in [5], [6], [7], which covers a preliminary trust framework, a computational model based on a numerical approach and a survey on existing computational models respectively.

There are several trust related frameworks can be observed the literature, like [8] and [9] based on privacy, [5] and [21] based on reputation, and [10] and [11] based on social relationships. On the other hand there are some

- *U. Jayasinghe, G. M. Lee, and Q. Shi are with Liverpool John Moores University, Liverpool L3 3AF, UK. E-mail: u.u.jayasinghe@2015.ljmu.ac.uk, {G.M.Lee, Q.Shi}@ljmu.ac.uk.*
- *T. Um is with Chosun University, Gwangju 61452, Korea. E-mail: twum@chosun.ac.kr.*

Manuscript received 15 Oct. 2017; revised 4 May 2018; accepted 15 May 2018. Date of publication 24 May 2018; date of current version 6 Mar. 2019. (Coressponding author: Gyu Myoung Lee.)
Recommended for acceptance by M. Qiu and S.-Y. Kung.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSUSC.2018.2839623

frameworks which are aiming particular application area like ad-hoc networks [12], P2P [13] and social networks [14]. However, these approaches lack generality in terms of application domain and target area. Therefore, it is essential that trust mechanisms are designed and developed to look ahead to the future where many individual objects are interconnected with new vulnerabilities possibly introduced in heterogeneous systems and application domains. To realize this in a rational manner, a two-step process is followed. As the first step, a novel framework is proposed that defines trust metrics (TMs) under three categories: "Knowledge", "Experience", and "Reputation" which represent all aspects of trust in any system. Then as the second step, the trust attributes (TAs), which represent major TMs, can be identified, depending on the application area and methods, which can assess them. The obvious benefit is that experts and systems can work on each individual TM, depending on their expertise areas and compose them later for a more complete solution rather than proposing individual pieces of inventions, which are less practical in real world scenarios.

As our approach here is more concentrated on numerical aspects, the focus of this paper is on generating numerically measurable values by combining mathematical methods with intelligent Machine Learning (ML) techniques. The choice of the method depends on a balance of several factors like accuracy, computational resources, efficiency, availability of data, and urgency of the situation concerned.

This paper is an extension of our previous work [6] in terms of architectural design model, feature extraction methodologies, and intelligent algorithms to analyse features and autonomically assess a trust value without human intervention. The major new contributions of this paper are to: (i) present a comprehensive trust framework model which specifies the formation of trust from raw data to a final trust value, (ii) offer an analytical approach to assess the data and evaluate each individual trust feature, (iii) present a clustering algorithm to label the extracted trust features, (iv) propose an intelligent model based on a multi-class classification algorithm to combine measured TMs to formulate a trust assessment model, and (v) evaluate the effectiveness of the findings in a simulative environment.

To the best of our knowledge, we are the first to propose a trust assessment scheme in the above order over multiple features of a real data set. It composes both numerical and machine learning concepts in addition to the novel framework proposed, which encourage existing systems to adapt these definitions and concepts to effectively collaborate and design systems that are more robust in future.

Further, we have compared and proved in later sections that the proposed algorithm shows 2 percent improvement in contrast to previous algorithms [15], [16]. In addition, the algorithm is capable of adapting to the changes of the interactions over time and gaining a more powerful insight compared to the traditional methods like the liner aggregation in which behaviors of the objects are believed in such a way that they would act in the same manner as before in future. This shows a prominent feature of our algorithm towards designing an autonomous system that is capable of assessing trust dynamically without external interventions but intelligent enough to predict future misbehaviors. Moreover, the algorithm is not limited to classify the trustworthiness based on the proposed features only but open to accept any number of features. Hence, the algorithm is essentially a generalized one and can apply in any use case from smart home to cross border application domains without any restrictions.

The remainder of the paper is organized as follows. In Section 2, we survey related contributions on trust modeling, management, and computation methods. Following that, the design principles of a trust framework are defined in Section 3, which provides a foundation for the work, proposed. Section 4 discusses a basic feature extraction methodology for a genuine data set according to the IoT concepts. Based on this methodology, the development of a ML based algorithm is presented in Section 5. The numerical model and algorithms are tested in a simulation environment elaborated in Section 6. The simulation results are discussed in Section 7. Finally, Section 8 concludes the paper and outlines our future work.

2 Related Approaches

Trust management technologies have been widely investigated in many fields including economics, sociology, and computer science [4], [17], [18]. Current research on trust management systems in computer science is often focused on solving security and privacy related issues. For example, trust management systems established on privacy policies are presented in [8], and [9]. A survey on trust and reputation systems based on ad-hoc networks is presented in [12]. They specifically discuss architectures, TAs, and scopes of a trust management system for such networks. Momani et al. [19] argue the difference between trust and security in wireless sensor networks (WSN). Furthermore, authors in [13] dispute a decentralized trust management platform for peer-to-peer (P2P) applications. They present an innovative approach to classifying trust based on credential and policy, reputation, and social network information.

On the issue of trust computational methods, authors in [7], [20] explain several trust evaluation schemes based on the concepts of network architecture, policy, reputation, and hybrid methods. Methods based on network architecture use some structural information like in-degree, out-degree and page rank concepts as in [5], [21] to extract some trust related properties like reputation. Basically, policy based mechanisms are used to estimate whether an object is trustworthy, depending on a set of predefined rules or credentials as in [22], [23]. Reputation systems keep a track of the status of interactions and behaviors in order to make a trust decision, such as those used by eBay [10] and KeyNote [11].

On the other hand, social interactions among objects disclose the valuable information of trust in analogy to the sociology concept of human interactions based on trust relationships. In this regard, authors in [14] and [24] have developed a social model of cyber objects corresponding to their owner's social behavior. In such models objects interact with each other based on their trust relationships and reveal many information in terms of trust as described in [25], [26]. Moreover, [27] and [28] discuss about trust assessment of a social network based on concepts like a community of interest, friendship, followers as well as frequency, duration and behavior of the objects. In a similar manner, authors in [29]

and [30] present a computational model for trust based on similarity, information reliability, and social opinions.

However, the influence of a particular TA on trust is determined by a weighting factor, but the assessment of a proper weightage is a complex task due to the fact that trust is a varying quantity which depends on many factors, e.g., expectations of a trustor, time, context, etc. Thus, schemes that are more intelligent are required to find these weighting factors and a threshold that defines a trustworthy boundary. Authors in [8], [31] and [32] investigate more innovative models and solutions for privacy, security and data integrity based on statistical and deep learning concepts. Moreover, authors in [33] and [34] propose a regression based model which compares the variation of trustworthiness with respect to trust features in mobile ad-hoc networks (MANET) and WSN. However, they have investigated a limited number of trust features, which only represent the system level information like packet forwarding ratio, Quality of Service, energy-sensitivity, capability-limitation, and profit-awareness. This motivates us to present a generic trust framework that represents features from both social level as well as system level data.

Recently, authors in [35], [36] and [37] present several trust management frameworks based on reinforcement learning and multiclass classification techniques which lay the foundation for the algorithms considered in this work. Even though these research achievements show some prominent results by applying ML techniques, they still lack the potential of being a generic algorithm that can be commonly applicable to any service domains without limiting to specific infrastructures like MANET, WSN, Underwater Acoustic Networks etc. In addition, they only consider quite limited as well as conventional factors like energy saved in a particular transaction, delay, intrusion sensitivity, throughput, etc. for the trust assessment process. Moreover, they are missing the information about extracting social features, an intelligent labeling method and a trust prediction technique.

3 Generic Trust Management Framework

Typically, trust can be seen as a metric used to evaluate social actors in consideration of mutual benefits, coordination, and cooperation. Actors continuously update their trust on others in response to perception fluctuations due to direct interactions and based on believes and opinions of others who are around. Trust is a crucial fact that affects the appetite of an object to consume a particular service or product offered by another object. This example can be seen in our everyday life where trust decisions are made. When purchasing a specific product, we may favor certain brands due to our trust that these brands will provide excellent quality compared to unknown brands. Trust in these brands may come from our previous experience in using these brands' products, from their reputations perceived by other people who bought their items and left opinions about those products, or from suggestions of your surroundings such as families and friends.

In analogy to above viewpoint, trust also affects the decision of an object to transact with another object in an IoT ecosystem in which all participating objects must take decisions based on trust to provide/receive services to/from other objects. However, building trust in IoT is much more difficult due to the inability of machine objects to generate perceptions about other objects around them like humans. Furthermore, it is difficult to quantify the exact trustworthiness value of an object with a high accuracy. This is even harder when each object has a different interpretation and perception of the term "trustworthy". Therefore, they may assign different trustworthiness values to a provider or a service. As an example, a service consumer object assigns "very trustworthy" to the provider for a specific transaction that it has performed. However, another consumer object might assign "untrustworthy" for a similar transaction from the same provider. These differences further increase the difficulty to determine the exact trustworthiness of a provider.

Therefore, it is essential to establish a generic framework which defines the blueprint of a trust management process while keeping in mind the diversity of trust features and hence the flexibility given to objects to choose best and practical measures. To clarify the ambiguities and definitions of trust, we use the following definitions in the context of a cyber world in this paper [3]:

Definition 1 (Trust). *It is a qualitative or quantitative property of a trustee, evaluated by a trustor as a measurable belief, in a subjective or objective manner, for a given task, in a specific context, for a specific time period.*

Definition 2 (Trust Model). *It comprises three TMs: Knowledge, Experience, and Reputation. Each TM is a collective representation of several TAs. Each TA represents the trustworthiness feature of a trustee.*

We use the term "trustor" to represent an object that is expected to initiate an interaction with another object and "trustee" as the second object that provides necessary information towards the trustor upon its request. The first thing that we want to emphasize in the definition of trust is the nature of the measurement that can take either a quantitative or a qualitative form. Apart from the well-known numerical measurements like similarity, accuracy, etc., qualitative properties like motivation, awareness, and commitment can also be used to judge certain situations in the process of trust based decision making. In addition, it is important to recognize trust as a belief even in the cyber world. That means, trust is a relative phenomenon and 100 percent belief is neither practical nor achievable in a diverse environment like IoT.

Moreover, the perception of trust can be either subjective or objective, depending on the requirement of the trustor and the availability of needed information. If the trustor wants TMs in a specific format that goes with the trustor's profile of interest, then the measurements can be characterized as subjective. On the other hand, objective measures can be described as TAs collected without any profile based filtering. Lastly, it is utmost important to define trust specifically for a particular task, context and time frame. For example, one might trust another for their cloud storage services but not for online streaming services, i.e., task dependent trust. Further, this trustworthiness relationship to obtain cloud services might be for a temporary duration and not for persistent time, i.e., time dependent trust. Moreover, a client might use different cloud services in different countries, as he does not trust the same provider globally, i.e.,

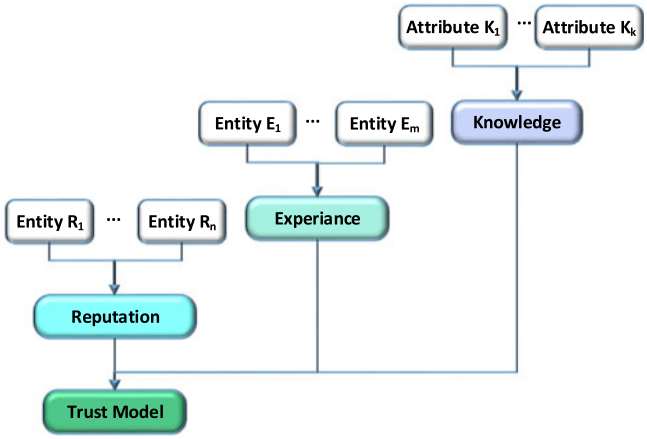

Fig. 1. Generic trust model: A prototype that explains the trust acquisition and evaluation process based on three TMs, knowledge, experience, and reputation as described in Sections 3.1 and 3.2.

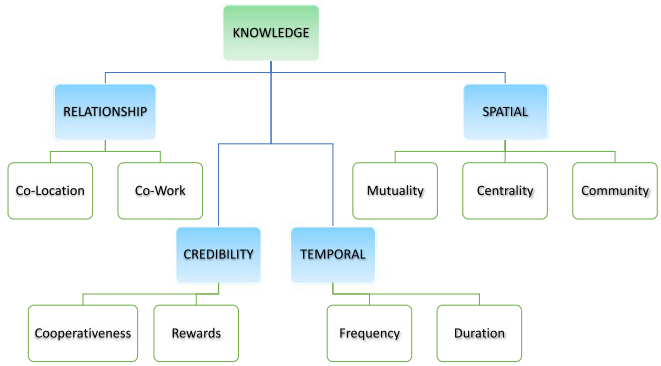

Fig. 2. Composition of knowledge: Describes the TAs that influence the evaluation of knowledge TM as explained in Section 3.1.

context dependent trust. Hence, trust is a variable in nature and hence cannot be assigned permanently to measure every task and every context of a specific actor or object. Further, we need to emphasize that trust is a relative quantity between two or more objects in opposite to a measurement of individual objects. Having stated the generic definition of trust, our next step is to define the course of trust acquisition, evaluation, and representation in an automatic or semi-automatic way in a computational setting, which we illustrate as the trust model in Fig. 1.

3.1 Knowledge Trust Metric

The knowledge TM covers all aspects of direct trust evaluations, which provide a perception about a trustee before an interaction. This is equivalent to analyzing the resume of a prospective candidate before hiring. To make this possible, it must provide relevant data to the trustor for its assessment. If a data feature can be represented using a quantitative measurement, then the result is a numerical value in a certain range. As an example, social relationships like co-location and co-work, credibility factors like cooperativeness, time dependent features like the frequency and duration of interactions, and spatial distribution of relevant trustees compared to the trustor can be used as direct trust measurements. The TAs, which we evaluate in this paper using ML techniques, are shown in Fig. 2.

The relationship TAs in Fig. 2 defines the mutual relationship between the trustor and a trustee. It is reasonable to assume that if two objects have a noble relationship between them, a higher trustworthiness can be expected between them. As an example, if the trustor and the trustee are operated closely by location such as looking for a parking lot near a supermarket, then both benefit (e.g., getting a vacant, closest, easily navigable parking lot) from their relationship based on location similarity that we have identified as co-location TA. Likewise, if the two objects are in a working relationship like car sharing in which one needs to provide a service and other needs to get the service, both can support each other via their co-work association.

Furthermore, it is important to maintain knowledge about the consistency of trustworthy service provisioning. We discuss properties related to this issue under credibility. The cooperativeness under credibility in Fig. 2 represents the level of social cooperation from the trustee to the trustor. The higher cooperativeness means the higher trust level in an IoT ecosystem. A user can evaluate the cooperativeness of others based on social ties and select socially cooperative users. Additionally, we have introduced a rewarding system in order to track the history of misbehavior situations or unsuitable reactions originated by the trustee. Rewarding TA can be used to either encourage or discourage further interactions with a particular trustee based on its past character.

To capture the significance of time related information to trustworthiness evaluation, TAs like the frequency and duration of the interactions can be used. It is logical to assume that the higher frequency and duration of interactions, the more trust is built up among the associating objects. On the contrary, the shorter time spent on each other, the less knowledge gathered on each other's behaviors and capabilities. As an example, in whitewashing attacks, a dishonest object can vanish for some time and rejoin the service in order to clear its reputation. However, if a trustor can keep a record of the consistency of the interested trustees then it can avoid such situations.

Moreover, in an IoT ecosystem, service provisioning (discover, manage and terminate) is based on its social relationships without solely depending on the underlying system level information. Therefore, it is vital to identify TAs, which determine the social proximity of the objects in collaboration. In this aspect, we identify three properties under spatial TAs in Fig. 2 as mutuality, centrality, and community of interest as governing features that define the social positioning of a trustee. Mutuality measures the degree of profile similarity between the trustee and trustor in resemblance to what is used in social networking. The community-interest represents whether the trustor and the trustee have a close relationship in terms of social communities, groups, and capabilities. Two objects with a degree of high community-interest have more opportunities in interacting with each other, and thus can result in a higher trust level. Centrality measures the importance of a trustee among other objects with respect to a particular task and context.

3.2 Experience and Reputation Trust Metrics

After acquiring enough evidences about trustees through the knowledge TM, the trustor can initiate collaborations with selected trustees based on the perception that the trustor has already obtained. However, the result of these interactions might differ from the perception and hence it is

critical to keep a record of each individual experience to be used in future interactions. For instance, experience might be a feedback from consumers after each transaction (as used in many e-Commerce systems), just a Boolean value $(0/1)$ indicating whether a service transaction successfully operates (as in some reputation-based trust systems), etc. Then, by accumulating these experiences over time in relation to the corresponding contexts, tasks and times, the trustor can build up additional intelligence compared to the knowledge TM.

To further enhance the perception of the trustor, other objects can share their experience in using the trustee, upon a request by the trustor, which we identify as reputation or the global opinion of the trustee. As an example, we have come up with a non-bias PageRank based model to calculate reputation values of trustees in a distributed network as in [5].

In summary, the experience TM is a personal observation considering only interactions from a trustor to a trustee, whereas the reputation TM reflects the global opinion of the trustee. However, the knowledge TM is the building block of both experience and reputation and hence the focus of this research is to generate quantitative results for the knowledge TM based on ML techniques.

4 Computational Model

Even though an IoT environment produces a large amount of data, it is questionable how much of them can be directly used for the trustworthy evaluation process. Therefore, it is vital to extract trust features by scanning social and system level interaction logs and store them in a data repository (DR) for further analysis. Hence, a numerical model that can extract basic features discussed in Section 3 is addressed here.

For that, we define the assessment of knowledge (K) towards an object j by an object i at time t as $K_{ij}^{x}(t)$, where x represents one of the features: Co-location relationship (CLR), Co-work relationship (CWR), Mutuality and Centrality (MC), Cooperativeness-Frequency-Duration (CFD), and Reward. Note that trust assessment is always between two or more objects.

4.1 Co-Location Relationship (CLR)

An IoT ecosystem enables users to share their resources, ideas, situations, and interested services with nearby devices. In such a situation, if both the trustor and the trustee are in close proximity and have subscribed to a DR in the platform, the trustor can conveniently get the required information from the selected trustee who is trustworthy in terms of the physical location compared to other objects far away from the scenario. However, in an IoT model, objects are always in relationship with their owner (Owner Object Relationship-OOR) and hence the static or dynamic nature of the OOR always affects the CLR [2]. In order to avoid objects leaving the physical location, a decision boundary based on the distance from the trustor (e.g., based on GPS data) and the time spent within this decision boundary are taken into consideration. Then the objects, which are within this distance boundary and exceed the minimum time threshold inside the region, are selected as prospective candidates for a trustee. Once the candidates are filtered, their CL relationship with the trustor can be calculated as follows:

$$K_{ij}^{\mathrm{CLR}}(t) = \frac{1}{\mathbf{dist}(i,j)} \frac{\mathrm{G_i G_j}}{\|\mathrm{G_i}\|\|\mathrm{G_j}\|}. \tag{1}$$

Here, g_i and g_j are the GPS coordinates of the trustor i and trustee j, respectively. The symbol "$\|.\|$" defines the norm of an element. The second term in (1) is the cosine similarity between the two objects and it is normalized by the geo distance factor $\mathbf{dist}(i,j)$ which can be calculated as in [38]. The application of the geo distance factor is important here as it provides a value with respect to an actual surface distance of the earth in contrast to a linear distance.

4.2 Co-work Relationship (CWR)

The objects that are collaborating in common IoT applications can be characterized as CWR. In such a situation, more focus would be on working relationship in a particular service domain rather than their physical proximity. To measure CWR as a numerical value, we compare the multicast interactions between a trustor and a trustee, as calculated below:

$$K_{ij}^{\mathrm{CWR}}(t) = \frac{\left|\mathbf{c}_{ij}^{\mathrm{MI}}\right|}{\left|\mathbf{c}_{j}^{\mathrm{MI}}\right|}, \tag{2}$$

where $\mathbf{c}_{\mathrm{ij}}^{\mathrm{MI}}$ is the vector of multicast interactions (MI) between trustor i and trustee j, and $\mathbf{c}_{j}^{\mathrm{MI}}$ is the vector of MI originated at j. The symbol "$|.|$" represents the determinant of a vector. $K_{ij}^{\mathrm{CWR}}(t)$ represents a relative measurement of shared multicast messages to total messages originated at the trustee.

4.3 Cooperativeness, Frequency, and Duration (CFD)

In a collaborative environment, it is important that every object execute its commitment to improve the level of the outcome of the whole service provision process. As an example, consider a malicious agent that provides fake ratings for a specific service. In this case, it is obvious that this agent deliberately tries to manipulate the genuineness of the information on the service and does not have any intention to use it. Therefore, the cooperativeness TA is vital to maintain the above-mentioned content stability and thereby to provide a trustworthy service to the trustor upon its request. Furthermore, it can be anticipated that the more frequent and longer the interactions among objects, the more collaboration from each party can be expected. Based on this, a numerical model for cooperativeness, frequency, and duration is derived.

Let us consider a set of interactions, $\mathrm{c}_1, \mathrm{c}_2 \ldots \mathrm{c}_n$ over some period in which the trustor is interested. A trust level between trustor i and trustee j is calculated below:

$$K_{ij}^{\mathrm{CFD}}(t) = \sum_{m=1}^{n} \frac{\mathrm{c}_m}{\mathrm{t}_m} E(c_m), \tag{3}$$

Here, n is the number of interactions, indicating how frequent they interact with each other. For the $m\mathrm{th}$ successful interaction, c_m is the length of an interaction between the trustor and the trustee, $\mathrm{t_m}$ is the total interaction length by the trustee. The factor $\mathrm{c}_m/\mathrm{t}_m$ assesses the duration property, in which the trustee interacts with the trustor, relative to the

total activity time of the trustee. $E(\mathrm{c}_m)$ is the binary entropy function which measures the balance in the interaction or the cooperativeness which can be calculated as follows [26]:

$$E\ (\mathrm{c}_m) = \ -p\ \log\ p - (1-p)\log\,(1-p), \tag{4}$$

where p is the fraction of the interactions between the trustor and the trustee. $E(\mathrm{c}_m)$ follows a binary distribution as stated in [39]. It is evident that the maximum entropy (i.e., $E(\mathrm{c}_m) = 1$) is reachable only when $p = 0.5$ that is 50 percent contribution from each party.

4.4 Reward System (RS)

An essential component of any service provisioning system needs to have a reward and punishing mechanism or a feedback model in order to assess the historical service experiences between a trustor and a trustee. It is always critical to maintain the social relationships at the maximum trustworthy level and hence we use the exponential downgrading formula shown in equation (5) for this purpose.

$$K_{ij}^{\mathrm{RS}}(t)\ = \frac{\|C\| - \left\|C_{\mathrm{p}}\right\|}{\|C\|}\ e^{\left(-\frac{\|C_{\mathrm{p}}\|}{\|C\|}\right)}. \tag{5}$$

Here, $||C||$ is the total number of interactions that have taken place during a period t, and $||C_{\mathrm{p}}||$ is the total number of unsuccessful or suspicious interactions. To punish misbehavior situations more severely, the slope of the distribution is increased, compared to the standard exponential distribution. Hence, a higher number of malicious interactions will result a lower reward value.

4.5 Mutuality and Centrality (MC)

In an IoT ecosystem, service discovery and provisioning largely depend on the social relationship among the participating objects. In this regard, the mutuality and the centrality TAs define the location of a trustee with respect to a trustor in a social world. On the other hand, it is very intuitive to assume that a higher number of mutual objects imply higher similarity between their social profiles. However, mutuality alone cannot be used as a TA due to the number of mutual friends being proportional to the number of friends of each individual object. That is, an object with a higher number of friends gets an additional advantage compared to an object that has recently joined the network but has higher trustworthiness. In order to avoid such circumstances, a relative measurement of mutuality compared to the total number of friends is considered. This is essentially the centrality property of the trustee and is calculated below:

$$K_{ij}^{\mathrm{MC}}(t)\ = \frac{\left|M_{ij}\right|}{|N_i|}. \tag{6}$$

where M_{ij} be the set of common friends between i and j, and N_i is the set of trustee's friends.

4.6 Community of Interest (CoI)

Objects in an IoT environment usually collaborate with at least one community. As an example, a person is registered as a frequent customer of a car sharing community while being a member of several other communities like online markets, social networking groups, etc. If another person is also a member of the car sharing community, this shows the resemblance of interest of both persons' interests. Similarly, if the trustor and the trustee share common interest groups, that is an indication of the degree of the common interest or similar capabilities of the trustee compared to the trustor. Mathematically, let us define M_{ij}^{coi} as the set of communities where both the trustor and the trustee are involved in, and N_{ij}^{coi} as the set of communities with each including the trustee as a member. Please note that both the trustor and the trustee can be a member of several communities and hence the trust level of the trustee based on CoI is calculated in (7).

$$K_{ij}^{\mathrm{CoI}}(t)\ = \frac{\left|M_{ij}^{\mathrm{CoI}}\right|}{\left|N_i^{\mathrm{CoI}}\right|}. \tag{7}$$

After the extraction of all the TAs using equations (1), (2), (3), (5), (6) and (7), the next step is to calculate the final trust value of the trustee. A well-known approach is to combine each TA through a linear equation with weighting factors as shown in (8).

$$\begin{aligned} K_{ij}\,(t) = \ & K_{ij}^{\mathrm{CLR}}(t) + \beta K_{ij}^{\mathrm{CWR}}(t) + K_{ij}^{\mathrm{CFD}}(t) \\ & + \varepsilon K_{ij}^{\mathrm{RS}}(t) + K_{ij}^{\mathrm{MC}}(t)\ + K_{ij}^{\mathrm{CoI}}(t). \end{aligned} \tag{8}$$

However, there are many drawbacks in this approach, including (i) lack of information and an infinite number of possibilities when it comes to estimate a weighting factor, (ii) unsuitability of a threshold based system to detect the trustworthiness of a particular trustee, and (iii) inability to identify which TA makes the most influence on the trust in a particular context. Thus, we will propose a new approach to the trust evaluation process in the next section.

5 Machine Learning Based Model

To overcome the weakness about the TA combination discussed in the previous section, we propose a ML based model to analyze the TAs extracted before and predict the trustworthiness of prospective transactions based on the trained model. In order to achieve this, we first use an unsupervised learning algorithm to identify two different clusters or labels, namely trustworthy and untrustworthy. The main reason to use the unsupervised learning over a supervised method is due to the fact of unavailability of a labeled training set based on trustworthiness relationships.

Then a multi-class classification technique like support vector machine (SVM) is used to train the ML model in order to identify the best threshold level that separates trustworthy interactions from others. In this research, our main objective is to differentiate malicious interactions from trustworthy interactions with maximum boundary separation and minimum outliers rather than classification itself. Therefore, it is not necessary to go for other algorithms like Random Forest, especially with a low dimensional dataset compared to its sample size used in this work. However, depending on the data set, dimensionality, number of classifications required and noise levels of the samples, a model comparison can be performed to find out the best possible algorithm for each individual case. A well-trained model like this can differentiate an incoming interaction between

two or more objects much more efficiently than linear weightage methods [15], [16] and is much more beneficial in the decision making process.

Let us define the number of features considered in the model as n and the length of the training set as m. We use the five features defined in Section 4, i.e., CWR, CFD, RS, MC and CoI to train our model. They are expressed as a feature matrix $X_{(j)}^{(i)}$ where i denotes the ith training sample and j signifies the jth feature among the n features. Moreover, the label of each training sample i is denoted by $\mathbf{y}^{(i)}$. However, training labels are not readily available and a method for the labeling will be discussed in the sub-section below. These allow us to identify each training set as $(X_{(j)}^{(i)}, \mathbf{y}^{(i)})$ for $i = 1, 2, \ldots, m$ and $\boldsymbol{j} = 1, 2, \ldots, n$. In the following sub sections, we break down our main algorithm in to two parts and explain it separately in Sections 5.1, and 5.2 respectively.

5.1 Algorithm I: Clustering and Labeling

In this section, we develop an algorithm based on the K-means clustering technique, which is specified in detail in Algorithm I, in order to group interactions based on the aforementioned features and thereby label each interaction as trustworthy or untrustworthy [40]. The K-means algorithm needs to define two initial conditions: number of clusters (k) and initial centroid positions (μ) that each interaction is assigned to. As there is no way to find out these values at the beginning of the algorithm, we randomly assign initial centroid locations for a range of cluster sizes, e.g., from $k = 1$ to $k = 5$. After that, steps 6-9 in Algorithm I are executed until the cluster points $''\mu''$ are not changing any further (i.e., until the convergence). Then, the Elbow method is used to find out the optimum cluster size which gives the lowest value for the K-mean cost function $\mathrm{J}(c, \mu)$ where c is the index of a cluster centroid and μ is the coordinates of cluster centroids with the dimension of k [40].

Algorithm I. Data Clustering and Labelling

1: **Input:** X **Output:** $\boldsymbol{y}$
2: Initialize cluster centroids $\mu_1, \mu_2, \ldots \mu_k \in \Re^{\mathrm{n}}$
3: **for** $k = 1$ to 5 do
4: Repeat until convergence: {
5: **for** $i = 1$ to m do
6: $c^{(i)} := \arg\ \min_{\mathrm{j}} ||\mathbf{X}^{(i)} - \mu_k||^2$
7: $\mu_{\boldsymbol{\kappa}}$: = Average of points assigned to cluster
8: k
9: **end for**
10: }
11: $\mathrm{J}^{(k)}\ (c, \mu) := \arg\ \min_k \mathrm{J}(c, \mu)$
12: **end for**
13: **Optimum** $k \leftarrow$ Elbow method $\leftarrow$ **plot** $\mathrm{J}^{(k)}$ vs k
14: **for** $i = 1$ to m do
15: **if** $c^{(i)}$ close to (0,0)
16: $y^{(i)} = 0$
17: **else if**
18: $y^{(i)} = 1$
19: **end if**

Note that initial inputs to the algorithm were normalized between [0, 1] in which "0" represents untrustworthiness and "1" the most trustworthiness. Hence, it is logical to label points close "0" as untrustworthy and vice versa after the clusterization step. Therefore, after the step 13 of the algorithm, the clusters close to the origin (i.e., all zero point) of the N dimensional space is marked as "0" or untrustworthy and the cluster away from the origin is identified as a trustworthy region. To check the influence of all n features at once, the Principal Component Analysis (PCA) algorithm based on Singular Value Decomposition (SVD) is applied to reduce the N dimensions to two dimensions for visualization purposes as below before applying the algorithm I [41]. Even though it is possible to extend Algorithm I for n features with regularization, we observe that the PCA method is more efficient with respect to computational complexity of unsupervised learning with regularization.

The first step of the PCA algorithm is to calculate the covariance matrix Σ that has the dimension of $n \times n$. In the step two principal components U and V are calculated using the SVD function, each having the same dimension as Σ [41]. As our intention, here to reduce the dimensions from five to two, dimensions (d) of the principal matrix U is set to two. Finally, step four calculates the two-dimensional feature vector Z in corresponding to five-dimension vector X.

Algorithm: Principal Component Analysis

1: **Compute dot product matrix:** $\boldsymbol{\Sigma = X^{\mathrm{T}} X}$
2: **Compute eigenvectors:** $[\boldsymbol{U, S, V}] = \mathrm{SVD}(\boldsymbol{X^{\mathrm{T}} X})$
3: **Specify the required dimension,** $d : \boldsymbol{U}_d = [\mathrm{u}_1, \ldots \mathrm{u}_d]$
4: **Compute** $d(= 2)$ **features:** $\boldsymbol{Z = U_d^{\mathrm{T}} X}$

5.2 Algorithm II: Classification Model

Having obtained the completed data set $(X_{(j)}^{(i)}, \boldsymbol{y}^{(i)})$ via Algorithm I, the next step is to train an algorithm based on a SVM technique which can identify the nonlinear boundaries of trustworthy and untrustworthy interactions. In order to obtain the maximum accuracy of the learning algorithm, the train set is divided into two parts in such a way that the training set occupies 80 percent of the data and 20 percent for the cross validation data set which is denoted as $(X_{\mathrm{val}}^{(i)}, \boldsymbol{y}^{(i)}{}_{\mathrm{val}})$ for $i = 1, 2, \ldots, \lfloor 0.2 * m \rfloor$ and $j = 1, 2, \ldots, n$. This is important to avoid overfitting data through the regularization parameter and variance.

In our Algorithm II, we use a Radial Basis Function Kernel (RBFK) due to the smaller number of features (n) compared to the training set samples (m) as the authors in [42] have claimed. Further, in order to optimize the computational resources, the LIBSVM library is used to run the RBFK kernel [43]. First, we run the RBFK kernel over multiple instances of regularization parameters and variances in order to find optimum parameters for the learning algorithm as shown in step 4-7 in Algorithm II. As an example both c and γ are varied as a geometric series (e.g., 0.01, 0.03, 0.09... 30) to save the time and computation resources. Then the parameters which give the minimum error in the prediction step are chosen as the optimum factors for the SVM model. Further, it is essential to improve the accuracy of the final ML model and suppress any noise generated by the previous clustering algorithm. Hence, we use regularization techniques to avoid such issues during the training process in Algorithm II.

Afterwards, the algorithm is trained for all the training data samples using the algorithm II and model parameters are recorded to estimate future trust values based on the

TABLE 1
Parameters of the Data Set

Parameter	Value	Parameter	Value
Nodes	76	Interactions	18226
Objects	5776	Communities	711
Messages	899	Message Type (UC/MC/BC)	266/57/576

incoming feature statistics. The function *svmtrain* is defined in the LIBSVM library and calculates the decision boundaries based on the RBFK kernel as per SVM technique. Similar to algorithm one, first we consider two trust features at a time and investigate the trust boundaries. After that, features, which are derived through PCA algorithm, are considered to investigate the effect of all five features on the trust boundaries.

Algorithm II. Classification Model

1: **Input:** $X, y, \boldsymbol{X}_{val}, \ \boldsymbol{y}_{val}$
2: **Output:** Weights and Decision boundary
3: //Find best parameters c and γ
4: **for** $c, \gamma = 0.01$ (multiple of 3) 30 do
5: $\quad$ model = svmtrain($\boldsymbol{y}, \boldsymbol{X}$, RBFK,$c, \gamma$)
6: $\quad$ prediction = svmpredict($y_{\mathbf{val}}, \ X_{\mathbf{val}}$, model)
7: $\quad$ error $[c, \gamma]$ = predictions $\neq y_{\text{val}}$
8: **end for**
9: **Choose** $c, \gamma \leftarrow$ **minimum** $[error]$
10: [***weights***, ***accuracy***, ***decision values***]
11: $\quad$ = svmtrain($\boldsymbol{y}, X$, RBFK,c, γ)

6 Simulation Setup

To extract the aforementioned trust features to be used in the ML algorithms, we would need traces of many objects, which are not available now for IoT. Hence, we have used traces taken at the SIGCOMM-2009 conference which is available in CRAWDAD [44], [45]. These traces contain the information on device proximity, activity logs, friendship information, interested groups, application level message logs, and data layer transmission logs. We map the information to match with the IoT concepts described in [9]. In other words, we define a set of features, CWR, CFD, RS, MC and CoI, related to IoT based on raw data found in the data set. Therefore, our experiment can be re-applied with any real world IoT data set for further experiments without any ambiguity. This leads to the parameter settings and scenario of our simulation, as detailed in Table 1. Among 76 nodes, each pair of them (Trustor and Trustee) with at least a single interaction between them are considered as objects to match with the IoT concepts.

After obtaining the trust feature vector X_j for each node pair, they are organized as in (9) to generate the m training samples. We have deliberately omitted the results from CLR as the data set itself was obtained from a very close proximity and it is not meaningful to test location-based trust in this scenario. The dimension of the training sample matrix is in order of $\mathrm{m} \times \mathrm{n}$ where $\mathrm{m} = 5776$ (node pairs) and $\mathrm{n} = 5$. The notation $[.]^{\mathrm{T}}$ is used to denote the transpose of a vector and has the dimension of $\mathrm{m} \times 1$. Note that feature normalization is not required here as each feature value is in the range of 0 and 1.

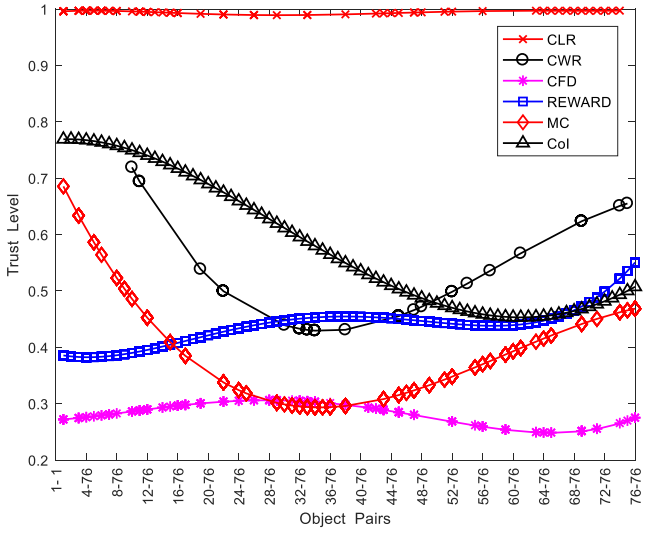

Fig. 3. Distribution of trustworthiness with respect to each feature. A 5th order polynomial is used to fit the data, distributed between 0 and 1.

$$[\boldsymbol{X}]_{m\times n} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ [CWR]^{\mathrm{T}} & [CFD]^{\mathrm{T}} & [RS]^{\mathrm{T}} & [MC]^{\mathrm{T}} & [CoI]^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \quad (9)$$

For the multiclass calcification problem, 4620 samples (i.e., 80 percent of the total samples) are chosen as the training set, and 1156 samples (i.e., around 20 percent of the total data set) are used as cross validation samples to avoid the data-overfitting problem.

For both ML experiments here, two features out of five are selected at a time for the sake of demonstration purposes, as it is not feasible to show a five-dimension vector. However, it is critical to analyze five features at the same time and evaluate their influence on the final trust value. Therefore, we then consider all the five features together and generate numerical results. However, to demonstrate the results, the PCA method is used to reduce the dimensions from five to two and generate the graphical results [46]. Note that PCA not only simplifies the visualization but also the algorithm complexity that make our model more practical in the case of a large number of features even though we use around five dimensions in this research to prove the effectiveness of our model in trust evaluation. Here, feature normalization is used to bring the new data samples, obtained through PCA, in the range of zero and one. The experiment is carried out on a PC which consists of 8 CPUs (Intel Core i7-2600, 3.4GHz) and 8GB RAM.

7 Performance Evaluation and Discussions

In this section, we present the simulation performance of our models in Sections 4 and 5. The simulation complexity is based on the number of interactions among objects and the number of nodes. For our feature extraction model, around 18000 interactions are used to generate each feature, and for the ML models, 5776 training samples are used.

7.1 Feature Extraction

Simulation results based on the numerical models defined in Section 4 are shown in Figs. 3 and 4. It is noticeable that the

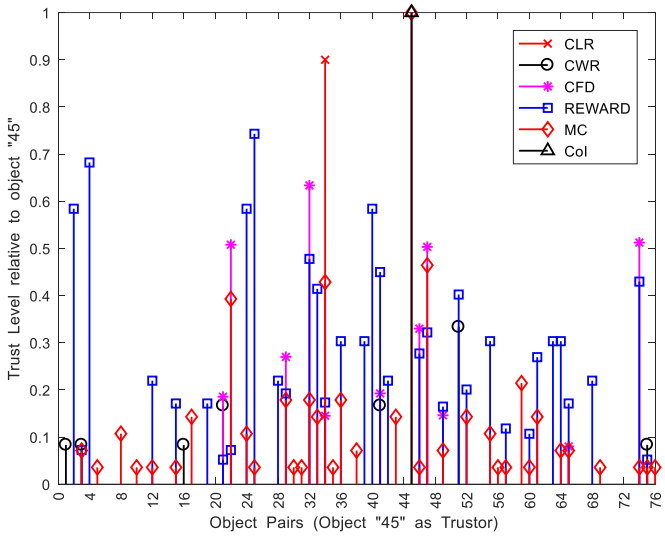

Fig. 4. Distribution of trustworthiness relative to object "45".

distribution of trustworthiness values in the CLR feature is close to "1" as shown in Fig. 3 due to the fact that the data is collected from the devices which are closely speeded. Note that only a fraction of object pairs have CLR association among 5776 objects as the data points represent those which have at least one transaction. Moreover, the trust values are normalized to fall between "0" and "1". One represents 100 percent trustworthy interactions and zero denotes untrustworthy interactions.

On the other hand, the distribution of CWR associations shown in Fig. 3 shows weaker associations compared to the CLR case even though they closely work together. Dissimilar intentions of each node can be one of the reasons that resulted in this kind of behavior. Further, the variation of trustworthiness values with respect to their cooperativeness, frequency and duration of the interactions is distributed towards the lower end of the graph as often radio frequency (RF) communications are limited to asymmetric type interactions as well as message exchanges of short durations. However, trust values based on the CoI and Centrality are distributed in the range of 0.3 to 0.8 in the figure showing some amount of profile similarity among the nodes. Furthermore, rewarding values given to each interaction are biased toward the lower end of the scale. This is mainly due to the unsuccessful or ill behaviors caused in the past interactions.

Similarly, Fig. 4 shows the distribution of trustworthiness of each object (Trustees) with respect to one specific object (Trustor). We have chosen object "45" randomly in order to

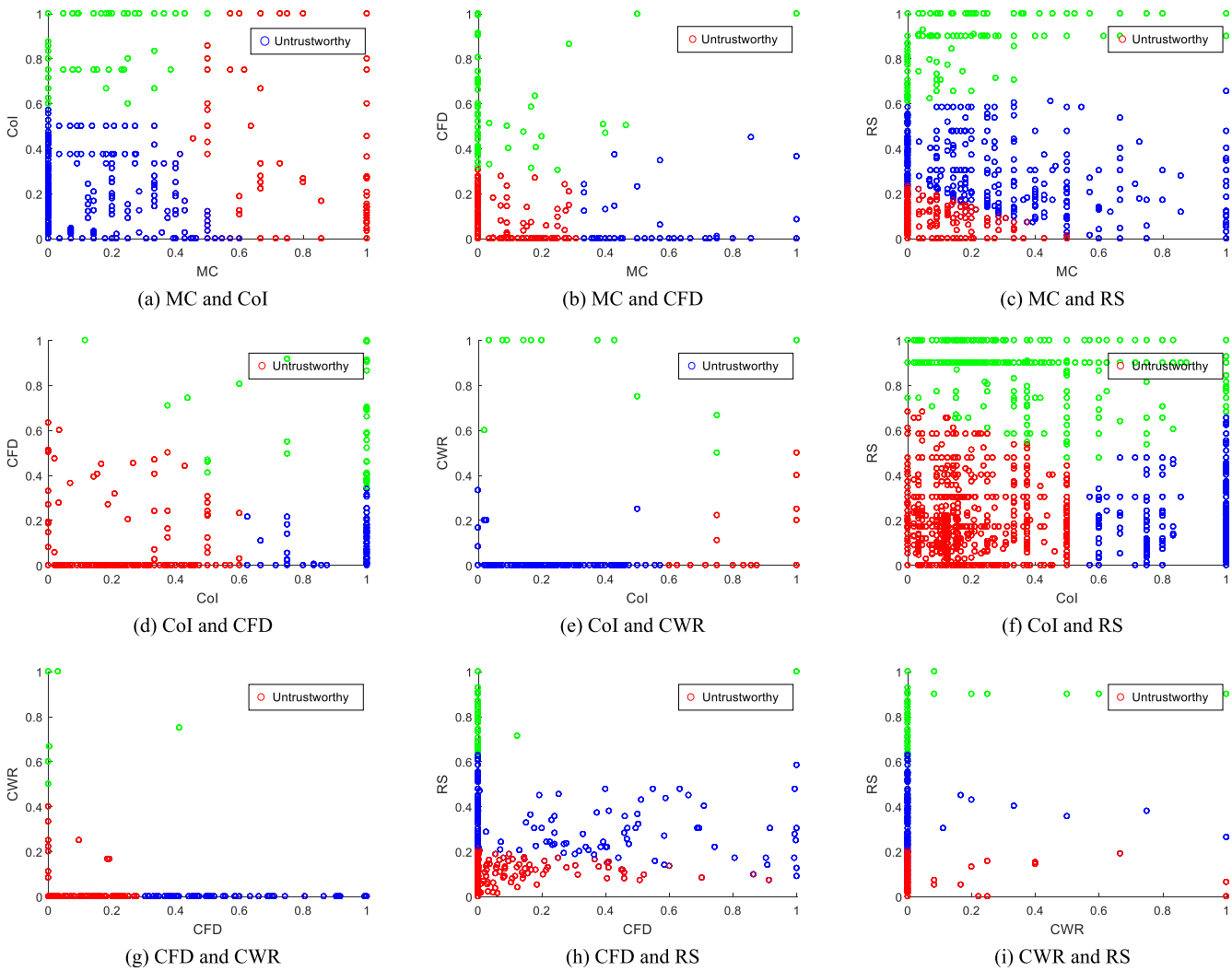

Fig. 5. K-means clustering on different pairs of features.

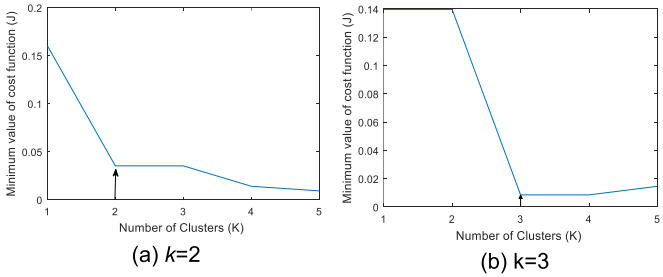

Fig. 6. Elbow method: To decide the optimum number of clusters-*k*.

generate these results. This figure clearly shows the interpretation of trustors view on other adjacent objects with respect to the features we have discussed in Section 4. As an example, trustee object "34" shows high co-location relationship with the trustor compared to other features while MC, CFD and Reward is around 0.4, 0.15 and 0.16 respectively. Therefore, it is possible that the trustor will engage in location-based services with the trustee in future interactions but limit its interactions related to collaborative services, as the MC and CFD values are low.

7.2 Algorithm I: Clustering and Labeling

With the successful abstraction of the trust features, the next step is to investigate how to combine each of them to generate a final trust value. To filter out most trustworthiness interactions from untrustworthy interactions, the algorithm explained in Section 5 is applied and the results obtained are shown in Fig. 5. In order to decide the optimum number of clusters, the Elbow method is used as shown in Fig. 6. In certain feature combinations, the algorithm is capable to categorize interactions into three groups as trustworthy, neutral, and untrustworthy. Instances where the Elbow method gives $\mathrm{K} = 3$ represent such situations. The results clearly shows the boundaries of separation from the untrustworthy interactions as marked in Fig. 5.

As an example, let us consider Fig. 5a that shows the distribution of trust values compared to centrality and community interest. It can be observed that the region above $\mathrm{MC} = 0.6$ and $\mathrm{CoI} = 0.6$ is the trustworthy region with respect to these two features. Similarly, Figs. 5b to 5g show a clear boundary between the trustworthy and untrustworthy regions. However, Figs. 5h and 5i show slightly different results compared to others. In both figures, the trustworthiness boundaries are learned with one

common feature: the reputation. From Figs. 5h and 5i, it is noticeable that the algorithm finds a lower trust value when the reputation value is low, even with a higher trustworthiness value of CFD or CWR. This is one of the interesting results as reputation is one of the critical factors when it comes to the trustworthiness evaluation process.

Note that we first run the algorithm pairwise to generate visual results and then combine all five features to find out the trustworthy region as shown in Fig. 7 where PCA is used to reduce the feature dimensions from 5D to 2D for the sake of visualizing the results. To bring the new dimensions into the range of 0 and 1, feature normalization is implemented. It can be clearly observed that values over around 0.5 on the 1st dimension and values over around 0.7 on the 2nd dimension show the boundary between trustworthy and untrustworthy interactions.

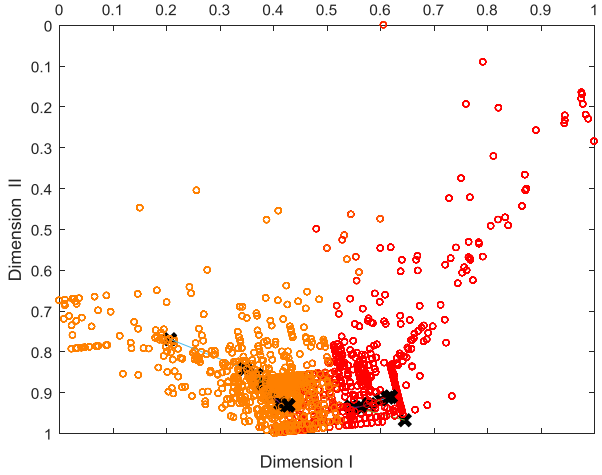

Fig. 7. Application of Algorithm I on features obtained via PCA.

7.3 Algorithm II: Classification Model

Having investigated which interactions belongs to the trustworthy region, we have used this information to label the data set. As an example, let's consider the same case in Fig. 5a. The points around the cluster centroid of the untrustworthy region are labeled as untrustworthy or "0" in the label vector "y", whereas the points outside the untrustworthy centroid are labeled as trustworthy or "1".

Then, with the labeled data, we train a model that can clearly identify whether incoming interactions are trustworthy. To estimate the optimum boundary, it is important to calculate the best regularization parameters "C" and "gamma" for each scenario mentioned above to avoid the data overfitting. For that, we have used part of the training samples as a cross validation set and the results obtained via the trained model are shown in Fig. 8 that clearly illustrates the decision boundary between the trustworthy and untrustworthy regions.

Furthermore, Fig. 9 shows the result after applying the dimensionality reduction for all five features. For instance, let us consider Fig. 8a in which the CoI and MC are in consideration. Now it is a matter of applying this model to the new data stream to distinguish which interactions fall into the trustworthy region and vice versa without any weight or threshold calculation. This not only reduces the calculation complexity and redundant work but also saves the processing time.

With these proven results, it is evident now that the system does not need to rely on conventional weighting factors and thresholds to decide the region of trustworthiness. However, the main assumption of this research is the centralized nature of the trust computation platform. Particularly, we assume that every object in consideration is subscribed to a centralized DR for publishing its data so that the trust computational platform can access the data, train a model, and publish the trust values back into the DR, which can be used by trustors.

To demonstrate the effectiveness of our proposed method over the most common methods like the liner aggregation of TAs, a confusion matrix method is considered. Classification accuracy often gives misleading results and hides the details needed to diagnose the performance of the model especially when the number of observations in each class varies as in our data set. On the other hand, the

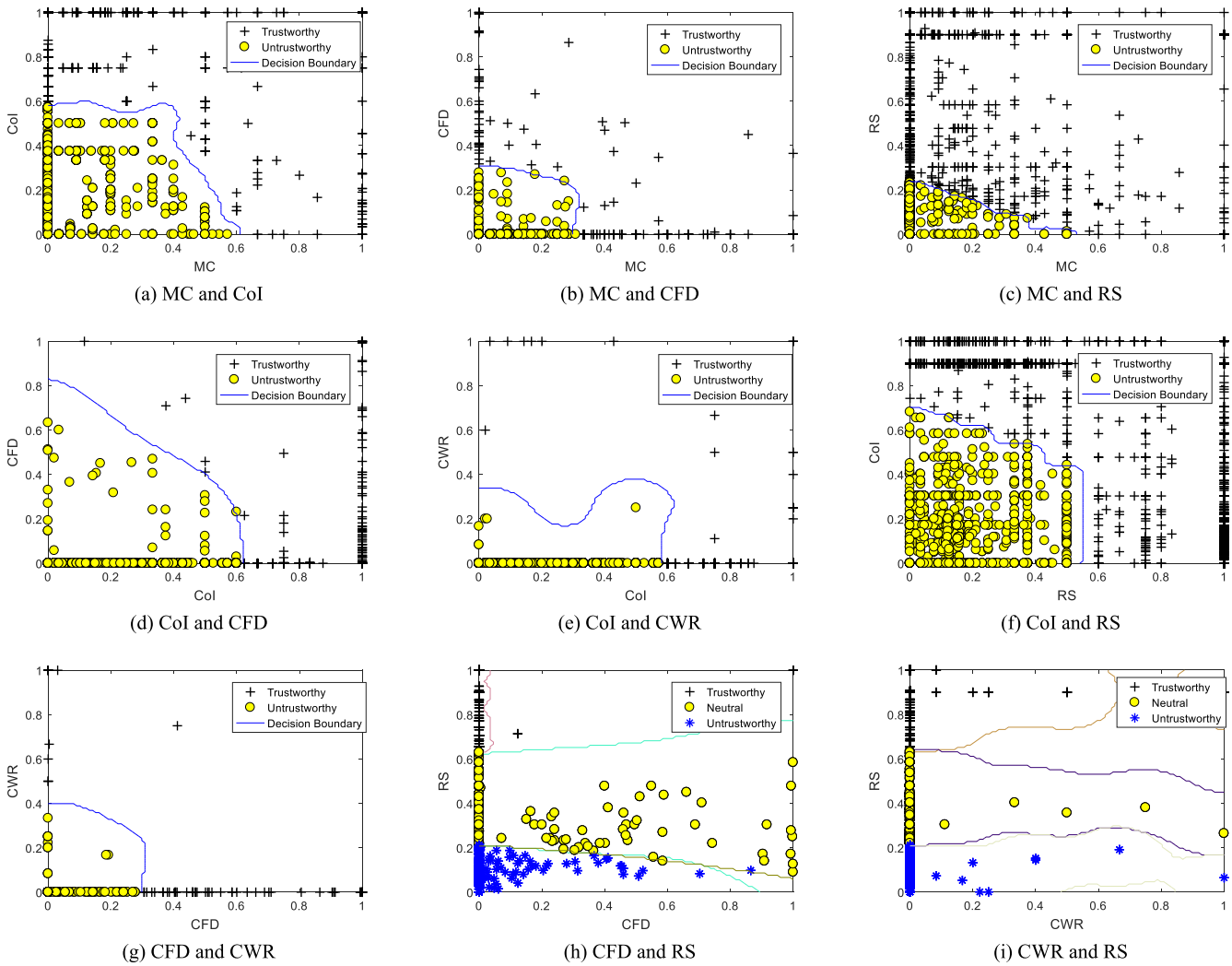

Fig. 8. Application of Algorithm II on different pairs of features.

confusion matrix shows at which point the algorithm makes errors or gets confused and importantly the types of errors made, which is critical for the investigation of algorithm applicability over expected results. For the comparison, we consider liner algorithms described in [15], [16] and a nonlinear algorithm described in [47]. The obtained results are shown in Table 2. Based on the results from Table 2, parameters that define the performance of each algorithm is shown in the Table 3.

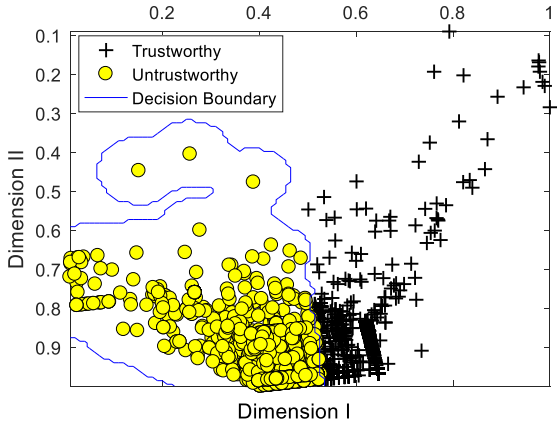

Fig. 9. Application of Algorithm II on all features obtain via PCA.

In classification, Recall gives an important insight about classification performance relative to the number of wrong predictions. According to our simulation results, the proposed algorithm shows 100 percent Recall or true positive rate (TPR) compared to 98.13 percent by the linear methods. As the data set is relatively small, 2 percent performance improvement in the proposed algorithm will be very critical in real world application deployment where billions of transactions happen in each second. This is again confirmed by the false negative rate (FNR) where the proposed algorithm shows 0 percent false negative predictions in comparison with 1.8 percent false predictions by liner methods. Note that TPR is similar in both proposed and nonlinear

TABLE 2
Algorithm Comparison with Confusion Matrix

		Trustworthy	Untrustworthy
Trustworthiness Prediction	Proposed	105	12
	Linear	105	2
	Nonlinear	105	19
Untrustworthiness Prediction	Proposed	0	2862
	Linear	2	2874
	Nonlinear	0	2855

TABLE 3
Parameters Derived from Confusion Matrix

	TPR/Recall	FPR	Precision
Proposed	1	0.004175	0.897436
Linear	0.981308	0.000695	0.981308
Nonlinear	1	0.006611	0.846774
Proposed	0	0.995825	
Linear	0.018692	0.999305	
Nonlinear	0	0.993389	
	FNR	**TNR**	

methods, as the nonlinear method only replaces the second part of the proposed algorithm. But, the proposed method outperforms the nonlinear approach as it gives a lower false positive rate (FPR) and a higher true negative rate (TNR) in contrast to the logistic regression, indicating that the proposed method shows compelling performance against untrustworthy objects.

Further, there are infinite possibilities when aggregating multiple TAs using a linear weighted summation method. However, in this comparison, the same weighting factors given by the clusterization algorithm are used in the liner algorithm to calculate the final score. Due to this reason, both proposed and logistic regression methods give a comparatively low score in contrast to the linear method. However, in realistic case, it is difficult to estimate these weighting factors without a proper clusterization algorithm as discuss in this work and hence precision will severely degrade compared to our proposed method. On the other hand, the regularization factor used to manage the over fitted data and the optimization algorithm used to find the optimum parameters for the features could have a significant effect on this cause. Thus, the precision of both models can be increased by observing the learning curve while tweaking this regularization factor depending on the data set and using advance methods of optimization as described in [48], [49].

Moreover, the algorithms described in this paper can be clustered so that the end devices can perform a fraction of the analysis and obtain the same results as before. This is quite beneficial in an environment like IoT where scalability and collaboration are prominent factors. To establish a distributed platform and address scalability issues, methods like map-reduction and data parallelism will be considered as strong candidate technologies in our future work [50].

8 Conclusion and Future Work

In this paper, a novel algorithm is proposed as opposed to traditional weighted summations to determine whether an incoming interaction is trustworthy, based on several trust features corresponding to an IoT environment. First, we have presented a generic trust computational model and a feature extraction method that can be applied to any service scenarios in IoT. Then a method for labeling the data depending on their trustworthiness is realized based on unsupervised learning techniques, which is the vital first step for any system to identify which interactions are trustworthy. Following this labeling process, a trust prediction model, which can correctly identify the trust boundaries of any interactions and learn the best parameters to combine each TA to obtain a final trust value, is proposed based on the well-known SVM model. Our simulation results have shown promising outcomes including the ability and accuracy of the algorithm with respect to identifying trustworthiness interactions.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2018R1A2B2003774).

References

[1] F. Y. Wang, "The emergence of intelligent enterprises: From CPS to CPSS," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 85–88, Jul.-Aug. 2010.
[2] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (SIoT) – When social networks meet the internet of things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012.
[3] Overview of trust provisioning for information and communication technology infrastructures and services, ITU-T, Recommendation Y.3052, 2017.
[4] S. P. Marsh, "Formalising trust as a computational concept," Ph.D. dissertation, Dept. Comput. Sci. Math., Stirling Univ., Scotland, UK, 1994.
[5] U. Jayasinghe, N. B. Truong, G. M. Lee, and T.-W. Um, "RpR: A trust computation model for social internet of things," in *Proc. Smart World Congress Int. IEEE Conf. Ubiquitous Intell. Comput.*, 2016, pp. 930–937.
[6] U. Jayasinghe, H. W. Lee, and G. M. Lee,"A computational model to evaluate honesty in social internet of things," in *Proc. 32nd ACM SIGAPP Symp. Appl. Comput.*, 2017, pp. 1830–1835.
[7] N. B. Truong, U. Jayasinghe, T.-W. Um, and G. M. Lee, "A survey on trust computation in the internet of things," *J. Korean Inst. Commun. Inf. Sci.*, vol. 33, no. 2, pp. 10–27, 2016.
[8] F. Fei, S. Li, H. Dai, C. Hu, and W. Dou, "A K-anonymity based schema for location privacy preservation," *IEEE Trans. Sustainable Comput.*, to be published, doi: 10.1109/TSUSC.2017.2733018.
[9] T. Jim, "SD3: A trust management system with certified evaluation," in *Proc. IEEE Symp. Security Privacy*, 2001, pp. 106–115.
[10] L. Xiong, and L. Liu, "A reputation-based trust model for peer-to-peer e-commerce communities," in *Proc. IEEE Int. Conf. E-Commerce*, 2003, pp. 275–284.
[11] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The KeyNote trust-management system," in *Proc. Int. Workshop Security Protocols*, 1999, pp. 59–63.
[12] M. A. Azer, S. M. El-Kassas, A. W. F. Hassan, and M. S. El-Soudani, "A survey on trust and reputation schemes in ad hoc networks," in *Proc. 3rd Int. Conf. Availability Rel. Security*, 2008, pp. 881–886.
[13] G. Suryanarayana, and R. N. Taylor, "A survey of trust management and resource discovery technologies in peer-to-peer applications," Institute for Software Research, California Univ., Irvine, Tech. Rep. UCIISR- 04-6, 2004.
[14] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Proc. 3rd Int. Conf. Ubiquitous Comput.*, 2001, pp. 116–122.
[15] 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.
[16] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proc. Int. Workshop Self-Aware Internet Things*, 2012, pp. 1–6.
[17] F. Huang, "Building social trust: A human-capital approach," *J. Institutional Theoretical Economics JITE*, vol. 163, no. 4, pp. 552–573, 2007.
[18] G. Möllering, "The nature of trust: From Georg Simmel to a theory of expectation, interpretation and suspension," *Sociology*, vol. 35, no. 2, pp. 403–420, 2001.

[19] M. Momani and S. Challa, "Survey of trust models in different network domains," *Int. J. Ad Hoc Sensor Ubiq. Comput.*, vol. 1, no. 3, pp. 1–19, 2010.
[20] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1–33, 2013.
[21] Y. Zhang, H. Chen, and Z. Wu, "A social network-based trust model for the semantic web," in *Proc. Int. Conf. Autonom. Trusted Comput.*, 2006, pp. 183–192.
[22] R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett, "No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web," in *Proc. Eur. Semantic Web Symp.*, 2004, pp. 342–356.
[23] W. Nejdl, D. Olmedilla, and M. Winslett, "Peertrust: Automated trust negotiation for peers on the semantic web," in *Proc. Workshop Secure Data Manage.*, 2004, pp. 118–132.
[24] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the IoT," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 97–105, Jan. 2014.
[25] S. Nepal, W. Sherchan, and C. Paris, "Strust: A trust model for social networks," in *Proc. IEEE 10th Int. Conf. Trust Security Privacy Comput. Commun.*, 2011, pp. 841–846.
[26] S. Adali, R. Escriva, M. K. Goldberg, M. Hayvanovych, M. Magdon-Ismail, B. K. Szymanski, W. A. Wallace, and G. Williams, "Measuring behavioral trust in social networks," in *Proc. IEEE Int. Conf. Intell. Security Inf.*, 2010, pp. 150–152.
[27] Y. Hu, D. Wang, H. Zhong, and F. Wu, "SocialTrust: Enabling long-term social cooperation in p2p services," *Springer P2P Netw. Appl.*, vol. 7, no. 4, pp. 525–538, 2014.
[28] M. Nitti, R. Girau, L. Atzori, A. Lera, and G. Morabito, "A subjective model for trustworthiness evaluation in the social IoT," in *Proc. IEEE Int. Symp. Personal Indoor Mobile Radio Commun.*, 2013, pp. 18–23.
[29] J. Zhan and X. Fang, "A novel trust computing system for social networks," in *Proc. IEEE 3rd Int. Conf. Privacy Security Risk Trust (PASSAT) IEEE 3rd Int. Conf. Social Comput.*, 2011, pp. 1284–1289.
[30] G. Yin, F. Jiang, S. Cheng, X. Li, and X. He, "Autrust: A practical trust measurement for adjacent users in social networks," in *Proc. 2nd Int. Conf. Cloud Green Comput.*, 2012, pp. 360–367.
[31] F. Jiang, Y. Fu, B. B. Gupta, F. Lou, S. Rho, F. Meng, and Z. Tian, "Deep learning based multi-channel intelligent attack detection for data security," *IEEE Trans. Sustainable Comput.*, to be published, doi: 10.1109/TSUSC.2018.2793284.
[32] J. Shen, D. Liu, D. He, X. Huang, and Y. Xiang, "Algebraic signatures-based data integrity auditing for efficient data dynamics in cloud computing," *IEEE Trans. Sustainable Comput.*, to be published, doi: 10.1109/TSUSC.2017.2781232.
[33] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, "LogitTrust: A logit regression-based trust model for mobile ad hoc networks," in *Proc. 6th ASE Int. Conf. Privacy Security Risk Trust*, 2014, pp. 1–10.
[34] Z. Li, X. Li, V. Narasimhan, A. Nayak, and I. Stojmenovic, "Autoregression models for trust management in wireless ad hoc networks," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–5.
[35] F. Boustanifar and Z. Movahedi, "A trust-based offloading for mobile M2M communications," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput.*, 2016, pp. 1139–1143.
[36] W. Li, W. Meng, L.-F. Kwok, and H. Horace, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *J. Netw. Comput. Appl.*, vol. 77, pp. 135–145, 2017.
[37] A. Bolster and A. Marshall, "Analytical metric weight generation for multi-domain trust in autonomous underwater MANETs," in *Proc. IEEE 3rd Underwater Commun. Netw. Conf.*, 2016, pp. 1–5.
[38] C. Veness, "Calculate distance and bearing between two latitude/longitude points using haversine formula in Javascript," Movable type scripts. (2016). [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html
[39] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, United Kingdom: Cambridge Univ. Press, 2003.
[40] X. Amatriain and J. M. Pujol, "Data mining methods for recommender systems, " *Recommender Systems Handbook*, pp. 227–262, Boston, MA, USA: Springer, 2015.
[41] S. Zafeiriou, "Notes on implementation of component analysistechniques," 2015; [Online] Available: https://ibug.doc.ic.ac.uk/media/uploads/documents/notes_implementation_component_analysis.pdf.
[42] C. Hsu, C. Chang, and C. Lin, "A practical guide to support vector classification," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Tech. Rep. 05, 2003.
[43] C. Chih-Chung and L. Chih-Jen, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.
[44] A.-K. Pietil, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "MobiClique: Middleware for mobile social networking," in *Proc. 2nd ACM Workshop Online Social Netw.*, 2009, pp. 49–54.
[45] P. Anna-Kaisa and D. Christophe, "CRAWDAD dataset thlab/sigcomm2009," (2012). [Online] Available: https://crawdad.org/thlab/sigcomm2009/20120715
[46] I. Jolliffe, *Principal Component Analysis*, New York, NY, USA: Wiley Online Library, 2002.
[47] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, "Logittrust: A logit regression-based trust model for mobile ad hoc networks," in *Proc. 6th ASE Int. Conf. Privacy Security Risk Trust*, 2014, pp. 1–10.
[48] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv:1212.5701, 2012.
[49] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, no. 1, pp. 503–528, 1989.
[50] H.-C. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker, "Map-reduce-merge: simplified relational data processing on large clusters," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2007, pp. 1029–1040.

Upul Jayasinghe received the BSc degree in electronics and telecommunication engineering (first-class honors) from the University of Moratuwa, Sri Lanka, in 2010 and the MSc degree from the Asian Institute of Technology, Thailand, in 2013. He is working toward the PhD degree at Liverpool John Moores University, UK. He is the recipient of the A. B. Sharma Memorial Prize in recognition of having the best thesis from the fields of Information, communication, and telecommunication engineering from the Asian Institute of Technology, in 2013. He has worked as a researcher in the Centre for Wireless Communication, University of Oulu, Finland, and the Computer Communications and Applications Laboratory, EPFL, Switzerland. His research interests include IoT, data analytics, networking and security, wireless communication, and mobile networks. He is a student member of the IEEE.

Gyu Myoung Lee received the BS degree in electronic and electrical engineering from Hong Ik University, Seoul, Republic of Korea, in 1999 and the MS, and PhD degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2000 and 2007, respectively. He is currently a reader in the Department of Computer Science with Liverpool John Moores University, Liverpool (LJMU), UK. He is also with the KAIST Institute for IT Convergence, Daejeon, Republic of Korea, as an adjunct professor. His research interests include future networks, internet of things, multimedia services, and energy saving networks including smart grid. He has actively contributed to standardization in ITU-T as a rapporteur (currently Q16/13 and Q4/20), oneM2M, and IETF. He is also the chair of the ITU-T Focus Group on data processing and management to support IoT and Smart Cities & Communities. He is a senior member of the IEEE.

Tai-Won Um received the BS degree in electronic and electrical engineering from Hong Ik University, Seoul, Korea, in 1999, and the MS and PhD degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2000 and 2006, respectively. He is currently an associate professor with Chosun University, Gwangju, Korea. He has been actively participating in standardization meetings including ITU-T SG 13 (future networks including mobile, cloud computing, and NGN).

Qi Shi received the PhD degree in computing from the Dalian University of Technology, P.R. China. He is a professor in computer security and the director of the PROTECT Research Centre in the Department of Computer Science, Liverpool John Moores University (LJMU), UK. He worked as a research associate with the University of York, UK. He joined LJMU, working as a lecturer and then a reader before becoming a professor. He has many years research experience in a number of areas, e.g., computer networks and security, secure service composition, privacy-preserving data aggregation, cryptography, computer forensics, formal security models, and cloud security. He has published more than 200 papers in international conference proceedings and journals, and served on a number of conference IPCs and journal editorial boards. He has also played a key role in many funded research and development projects related to his research topics.