Trust2Vec: Large-Scale IoT Trust Management System based on Signed Network Embeddings

Sahraoui Dhelim, Nyothiri Aung, Tahar Kechadi, Huansheng Ning, Liming Chen and Abderrahmane Lakas

***Abstract*—A trust management system (TMS) is an integral component of any IoT network. A reliable trust management system must guarantee the network security, data integrity, and act as a referee that promotes legitimate devices, and punishes any malicious activities. Trust scores assigned by TMSs reflect devices' reputations, which can help predict the future behaviours of network entities and subsequently judge the reliability of different entities in IoT networks. Many TMSs have been proposed in the literature, these systems are designed for small-scale trust attacks, and can deal with attacks where a malicious device tries to undermine TMS by spreading fake trust reports. However, these systems are prone to large-scale trust attacks. To address this problem, in this paper, we propose a TMS for large-scale IoT systems called Trust2Vec, which can manage trust relationships in large-scale IoT systems and can mitigate large-scale trust attacks that are performed by hundreds of malicious devices. Trust2Vec leverages a random-walk network exploration algorithm that navigates the trust relationship among devices and computes trust network embeddings, which enables it to analyze the latent network structure of trust relationships, even if there is no direct trust rating between two malicious devices. To detect large-scale attacks, such as self-promoting and bad-mouthing, we propose a network embeddings community detection algorithm that detects and blocks communities of malicious nodes. The effectiveness of Trust2Vec is validated through large-scale IoT network simulation. The results show that Trust2Vec can achieve up to 94% mitigation rate in various network settings.**

***Index Terms*—IoT, trust management, network embedding, bad-mouthing, self-promoting, device trust.**

I. Introduction

THE wide deployment of Internet of Things (IoT) applications has created a large network of interconnected physical devices, as well as virtual entities, such as agents. Managing trust relationships among this huge number of IoT devices is an important part of IoT security. A trust management system is used to ensure network security and data integrity in IoT [1]. A Trust management system (TMS) can serve as a referee that promotes well-behaved entities and punishes malicious devices within the network. To do so, a TMS assigns a trust score for each entity in the network. A Trust score is a good indicator for predicting future behaviors of the network entities and subsequently judging the reliability of different entities in an IoT network. However, if malicious entities manage to alter the trust scores, the trust and reputation indicators might not reflect the genuine nature of network entities. Therefore, the TMS may mistakenly punish reliable entities and reward malicious entities. Furthermore, such a fake trust score could pose a serious threat to the functioning of the whole system and may enable network attackers to gain access to sensitive information [2]. Trust among IoT devices is usually measured and evaluated using two factors [3], namely direct trust and indirect trust. The former represents the personal experience of a given device with regards to other network entities, it is usually computed by rating the previous experience with these entities. The latter represents the reputation score of a device, it is computed by aggregating multiple ratings given by entities that interacted with the device.

The most known trust-based attacks are self-promoting attack [4] (also known as a good-mouthing attack) and bad-mouthing attack [5]. In self-promoting attacks, malicious devices attempt to illegally increase their trust scores (reputation). The attack could be conducted by two nodes, or by a large number of nodes that work together to achieve their malicious purpose. In the most basic form of a self-promoting attack, two nodes provide a false report for each other to promote themselves as trustworthy entities, hence increasing their trust scores (reputation). To mitigate the self-promoting attack, a TMS must keep track of all previously reported trust ratings, and detect and punish the entities that are involved in such malicious activities. In bad-mouthing attack, attackers usually give bad ratings to a victim entity in order to lower its trust score and destroy its reputation among other nodes. Figure 1 and Figure 2 show examples of self-promoting and bad-mouthing attacks. In these figures, the white circles denote normal entities, and the red circles denote malicious entities that perform an attack. A solid arrow represents a positive trust rating and a dashed arrow represents a negative trust rating. Figure 1 (a) illustrates an example of small-scale self-promoting, where two malicious nodes increase their trust scores by repeatedly giving each other positive ratings. Figure 1 (b) demonstrates that two malicious nodes undermine the reputation of a legitimate node by continuously giving it negative trust ratings. Such small-scale attacks can be easily mitigated by controlling the rating behaviors of each entity in the system. For example, to prevent self-promoting attacks, a TMS can limit the number of positive trust ratings that two entities are allowed to give to each other. Similarly, bad-

Sahraoui Dhelim, Nyothiri Aung and Tahar Kechadi are with the School of Computer Science, University College Dublin, Ireland.

Huansheng Ning is with School of Information Technology and Engineering, Jinzhong University, Shanxi, 030619, China

Huansheng Ning is also with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

Liming Chen is with the School of Computing, Ulster University, U.K.

Abderrahmane Lakas is with the College of Information Technology, United Arab Emirates University, UAE

Sahraoui Dhelim and Nyothiri Aung are co-first authors, they have contributed equally to this work

Corresponding author: Huansheng Ning (ninghuansheng@ustb.edu.cn).

mouthing attacks can be prevented by limiting the number of negative trust ratings that an entity can assign to another. However, things get more complicated when a group of entities is collectively involved in self-promoting and bad-mouthing attacks. For example, in Figure 2 (a) a group of malicious nodes increase their trust score by giving each other positive ratings without attracting any attention, achieve this in the way that each node gives no more than one positive rating to another node in the malicious group. Similarly, in Figure 2 (b) a group of malicious nodes performs bad-mouthing attacks against a normal node by targeting it with unfair ratings.

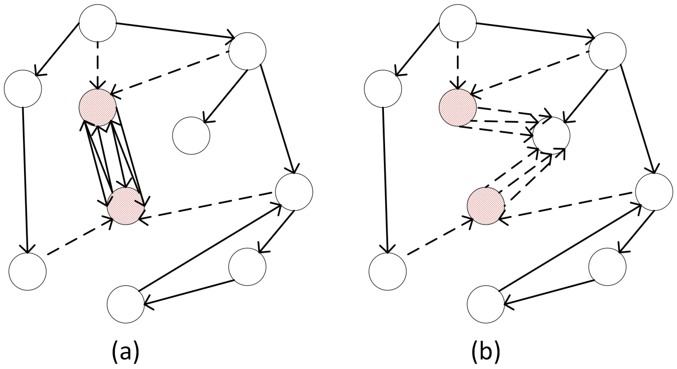


Fig. 1: (a) small-scale self-promoting attack. (b) small-scale bad-mouthing attack

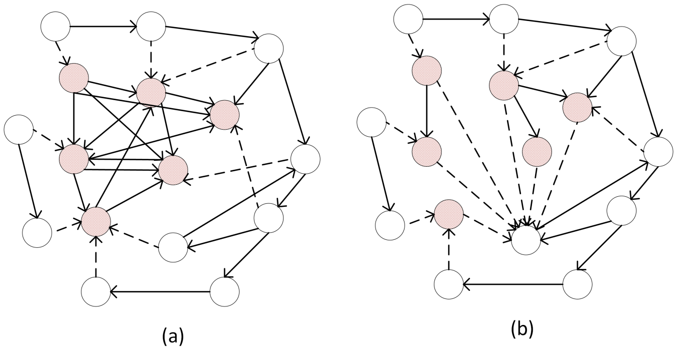


Fig. 2: (a) large-scale self-promoting attack. (b) large-scale bad-mouthing attack

While existing trust frameworks can mitigate small-scale trust-related attacks, managements frameworks that considered large-scale trust-related attacks have not been seen yet. On the other hand, network embedding algorithms have been proven effective when dealing with large-scale graphs and networks [6]. Therefore, in this paper, we propose a trust management framework, dubbed as Trust2Vec, for large-scale IoT systems, which can manage the trust of millions of IoT devices. Trust2Vec can mitigate large-scale trust attacks that are performed by hundreds of malicious nodes. Our contributions can be summarized as follows:

- Propose a trust management framework that can mitigate large-scale and small-scale trust-related attacks, such as self-promoting and bad-mouthing attacks.
- Develop a random-walk network algorithm that navigates the trust relationships among devices and computes trust network embeddings. The algorithm enables the proposed system to analyze the latent network structure of trust relationships.
- We developed a parallelization method for trust attack detection in large-scale IoT systems. Parallelizing made the TMS highly scalable and can manage a large number of network devices with less computational cost.

The rest of the paper is organized as follows: Section II reviews existing research about trust management in IoT. Section III describes the system design of the proposed trust management framework, and how Trust2Vec is used to detect trust-related attacks. Section IV presents the evaluation details and experiment results. We conclude the paper and outline future research directions in Section V.

II. Related Work

Trust management in IoT is a well-established research topic in the literature. Guo et al. [7] proposed a data collection method for IoT using UAV that uses a trust score to evaluate the reliability of data collection devices. They concluded that using trust as an evaluation metric for UAV data collection can significantly increase the data accuracy and reduce data collection costs. Similarly, Liang et al. [8] investigated the usage of trust management in UAV-assisted IoT. They proposed a trust evaluation scheme to identify the trust of the mobile vehicles by dispatching the UAV to obtain the trust messages directly from the selected devices as evidence. Kumar et al. [9] introduced a smart city networking architecture that leverages a trust computational module to distinguish unreliability and trustworthiness among smart city sensors and devices. Fang et al [10] proposed a trust management framework, in which the devices in the cluster start to detect the nearby devices within sensing range, compute their trust value, and report to a pre-elected cluster head. The latter calculates the aggregated trust score of each device in the cluster. The cluster head is periodically re-elected by the network devices with the cluster. Chen et al. [11] introduced IoT-HiTrust, a 3-tier cloud-cloudlet-device hierarchical trust-based service management protocol for large-scale mobile-cloud IoT systems. Their proposed trust model combines friendship similarity, and social contact similarity to compute the trust score of network devices. In their study, the trust score is represented as a random variable in the range of [0, 1] following the Beta (α, β) distribution. The numbers of positive and negative experiences of an IoT device are represented as binomial random variables. They computed the indirect trust as a weighted sum of service ratings reported by other IoT devices, such that trust reports of socially similar devices are prioritized. Bahutair et al. [12] introduced a generic trust management framework that can operate for crowdsourced IoT services. Their framework leverages a multi-perspective trust model that obtains the implicit features of crowd-sourced IoT services. Each entity is represented by a set of characteristics that contribute to the entity's influence on trust. The trust features are fed into a machine-learning algorithm that manages the trust model for crowdsourced services in an IoT network. Marche el al. [13] discussed possible trust attacks that can affect IoT networks, and introduced a trust management model that is able to overcome trust-related attacks. Specifically, they proposed a decentralized trust management model based on Machine Learning algorithms. The model utilized several parameters to compute three trust scores, namely the goodness, usefulness, and perseverance score. Their model uses these scores to detect malicious nodes performing trust-related attacks. Movahedi et al. [14] proposed T-D2D, a lightweight trust model that evaluates a network device's trust level using both short-term and long-term evaluation intervals to mitigate different types of trust-related attacks. T-D2D records marginal misbehaving over several successive time slots to reveal the nature of suspicious malicious nodes with a light misbehaving attitude. To mitigate bad-mouthing attackers, T-D2D does not rely on other nodes' recommendations in the case when the direct trust is not decisive. T-D2D evaluates the honesty of a recommender based on the correctness of its recommendations over time. Ben Abderrahim et al. [15] introduced DTMS-IoT, a Dirichlet-based trust management system for the IoT, which alleviated dishonest trust recommendations and related attacks by clustering devices using the k-means algorithm. DTMS-IoT detects IoT devices' malicious activities, which allows it to alleviate the effect of on-off attacks and dishonest recommendations. Liu et al. [16] introduced a semi-centralized TMS that leverage blockchain for single and multiple domains. The devices are connected in a centralized fashion and coordinated by a cloud server that manage the rating data ledger, to support cross-domain data exchange the server uses rotation consensus protocol. The proposed TMS aggregates both direct and indirect trust information to compute the trust values of IoT devices. Din et al. [17] introduced a trust framework for lightweight devices, which uses a centralized trust authority. The framework manages trust certificates that enable devices to exchange services without prior knowledge or performing trust computations. Trust between two devices is computed by direct observations in terms of delivery ratio, compatibility, and cooperativeness, while trust recommendations are utilized to determine trust in the case of indirect observations. Okuda et al. [18] proposed a random-walk community detection algorithm that clusters similar nodes. Nodes that frequently appear when traversing the network using finite-length random walk are judged to belong to the same community. Aung et al [19], [20] studied trust relationship among driverless cars in the context of vehicular ad-hoc networks (VANET) for route recommendations and path planning, and also trust-based content caching [21], [22]. Wu et al. [23], [24] proposed a deep-learning (DL)-based physical layer authentication scheme which exploits channel state information to enhance the security of mobile edge computing systems. Dhelim et al [25] studied the trust among social network users for personality-aware recommendation system, they concluded that recommendation accuracy can be significantly improved by adding social factors such as trust and personality traits. Wang et al [26] studied trust relationships in human-machine hybrid artificial intelligence. Similarly, Cai et al [27] suggested that trust can be established in human-robot interactions.

All the above-mentioned trust frameworks were designed to mitigate small-scale trust attacks, without consideration for large-scale trust attacks. That is due to the challenge of analysing a large number of IoT devices with limited computational power required to analyse the trust relationships. In our proposed system, we have considered both small-scale, as well as large-scale trust attacks. We have overcome the computational cost limit problem by analysing latent network embeddings of trust relationships among IoT devices.

III. System model

To detect and mitigate a trust-related attack, Trust2Vec will analyse the network structure of trust relationships among devices. The main phases of the attack detection process, as depicted in Figure 3 are: 1) determine device communities; 2) generate random walks within each local community, which yield the devices' trust network embeddings; 3) leverage trust relationship network embeddings to detect malicious device clusters.

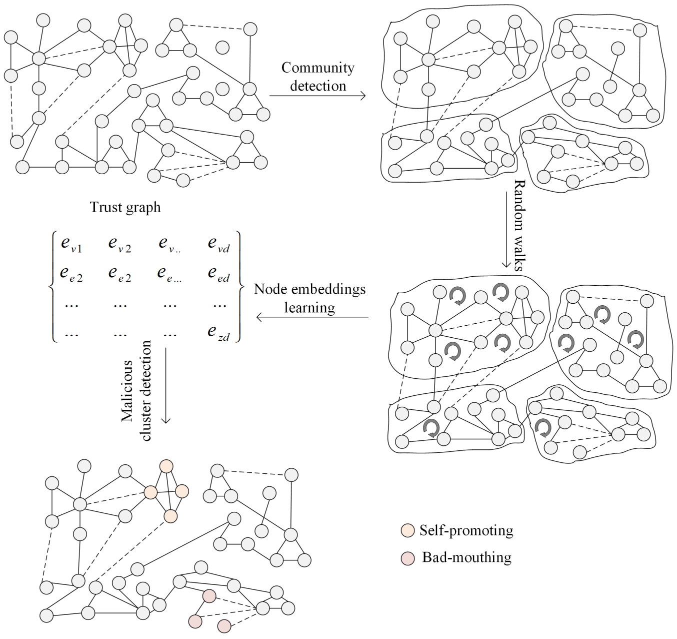


Fig. 3: Trust2Vec mitigation scheme phases

A. Community Detection

The network of devices can be structured as a graph $G = (D, T)$ that represents a signed trust network between devices, where the graph vertices represent the IoT devices; $D = \{d_1, d_2, \ldots, d_n\}$, and the edges represent the previous trust reports between graph nodes; $T = \{T^+, T^-\}$. An edge $t_{i,j} \in T^+$ denotes that device d_j is declared as trustworthy by device d_i, and $t_{i,j} \in T^-$ denotes that device d_j has been declared as untrustworthy by device d_i. Given the enormous size of IoT networks, which can have millions of devices, it is not viable to operate on the overall IoT network. However, the IoT network is generally easy to partition, as it is composed of smaller IoT subnetworks that are known as IoT units [1]. The edge density (trust relationships in our case) within these

IoT units is much higher than the edge between these Units. This is based on the observations that interactions between edge devices to perform edge computing tasks generate a high number of trust relationships as these devices work together within the same local network [28]. This allows us to detect the communities' boundaries without decreasing the resolution limit, an advantage that is not possible to get with other types of networks, such as social network graphs or user-products graphs [29], where the communities tend to overlap with each other [30].

B. Random Walks Generation

To compute the latent network structure, Trust2Vec generates random walks by navigating the network through random steps. The logic behind this approach is that the more we repeat these random walks from the same starting device with a fixed length, the more likely the walks will include nodes that are similar in terms of network proximity, either first-order proximity or second-order proximity nodes. Let $W_{d_i}(l)$ denotes the random walk starting from device d_i with a walk length l. A random walk $W_{d_i}(l)$ is a stochastic process composed of a chain of random variables $W_{d_i}(l) = \left\{w_{d_i}^1, w_{d_i}^2, \ldots, w_{d_i}^l\right\}$, where $w_{d_i}^n$ is the neighbor device that is randomly chosen among $w_{d_i}^{n-1}$ neighbors. Relying on random walks to compute the latent network structure is desirable for two reasons. Firstly, the computations of the random walks can be distributed among edge devices, hence the process is distributed and can be performed offline without relying on any remote cloud or server. Secondly, as the random walks are limited with walk length l, the newly added trust relationships among devices can be easily accommodated by regenerating random walks only for the updated trust relationships, without the need to recompute the random walks for the whole graph. To navigate the trust relationships among devices, Trust2Vec estimates the likelihood of observing a series of short random walks by adapting the same approach in natural language modeling, where the goal is to estimate the likelihood of a sentence being present in a corpus. The analogy here is to calculate the trust walk TW that starts from source device d_o, and estimate the probability of arriving at destination device d_d given the previously navigated devices through a random walk of length l:

$$TW\left(d_s, d_d, l\right) = Pr\left(d_d \mid \left(w_{d_s}^1, w_{d_s}^2, \ldots, w_{d_s}^{l-1}\right)\right) \quad (1)$$

C. Node Embeddings Learning

The objective is to learn the device's latent trust structure in the trust graph, not only the neighboring nodes that are a few hops away. Trust2Vec computes a device's trust network structure and represents it as a vector in low dimensional space. Formally, let $\phi : d \in D \longmapsto \mathbb{R}^{|D| \times z}$ be the mapping function that represents a device's latent trust structure for each device in D. Here z is the length of the vector in the lower dimension, such that $z \ll |D|$, hence it is easier to manipulate small vectors rather than the large adjacency matrix. The likelihood estimation is then represented as:

$$Pr\left(d_d \mid \left(\phi(w_{d_s}^1), \phi(w_{d_s}^2), \ldots, \phi(w_{d_s}^{l-1})\right)\right) \quad (2)$$

As the random walk length increases, it becomes computationally expensive to calculate the objective function. To overcome this problem, Trust2Vec leverages the skip-gram model [31], which inverses the problem by predicting the context given the missing word instead of predicting the word given the context. The skip-gram model maximizes the likelihood of any word to be observed in the current context without prior knowledge about current words. In the context of trust management, Trust2Vec target the following objective function:

$$\underset{\phi}{\operatorname{Min}} \left[-\log Pr\left(\left\{d_{i-w}, \ldots, d_{i-1}, d_{i+1}, \ldots, d_{i+w}\right\} \left|\phi(d_1)\right)\right)\right] \quad (3)$$

where w is the random walk window size.

Trust2Vec optimizes the function (3) to build the trust network representation for each device, hence capturing the latent similarity between network devices. Devices that have a similar trust network structure will have similar trust vectors in lower-dimensional space. Measuring the similarity of devices' representation in lower-dimensional space allows us to reveal the community membership, thus detecting malicious devices that perform trust-based attacks.

To computing devices' trust relationships and extract trust network embeddings, we developed Algorithm 1. The algorithm takes as an input the graph of devices in the studied edge environment, and generate the matrix $\phi \in \mathbb{R}^{|D| \times z}$ that represents the trust relationship network structure in lower dimensional space. Firstly, the algorithm generates the random walks of length l for λ times starting from each device d_x (line 3-5), and for each device within the ω hop away in the random walk $W_{\ d_x}$ apply the SkipGram model to map every device d_y to its representation vector $\phi(d_y) \in \mathbb{R}^z$. Given the low dimension representation, we are aiming to maximize the probability of the device neighbors in random walks (line 7). The posterior probability can be computed using basic classifiers such as logistic regression. However, this approach is not feasible as the number of devices increase, and become computational expensive to perform. To address this, we leverage hierarchical softmax [32], which maps all the network nodes (devices in our case) to a binary tree. In this way, the prediction problem is pivoted to maximizing the probability of path navigation from the root to the leaf of the tree that identify that device. In case a device d_k is defined by the sequence $\left(b_0, b_1, \ldots., b_{\lceil log[D] \rceil}\right)$ where b_0 is the root and $b_{\lceil log|D| \rceil} = d_k$ then eq(3) can be computed using binary classifier, which reduces the complexity of computing $Pr(d_k|\phi(d_y))$ from $O(|D|)$ to $O(log\,|D|)$ as showed bellow in eq(4)

$$\Pr\left(d_k \,\middle|\, \phi(d_y)\right) = \prod_{l=1}^{\lceil log|D| \rceil} Pr(d_k|\phi(d_y)) \quad (4)$$

To optimize eq(4), Trust2Vec utilizes Stochastic gradient descent (SGD). The derivatives are computed using the back-propagation algorithm, SGD learning rate is initialized as 2.5%

at the start of the training and then decreased linearly with the count of devices encountered so far.

Algorithm 1 Device_Trust_Embeddings

```
Input
graph $G = (D, T)$
Walk window size $\omega$
Low dimension vector size $z$
Random walks count per device $\lambda$
Random walks length $l$
Output
$\phi \in \mathbb{R}^{|D| \times z}$ Matrix of device trust embeddings

Sample_matrix($\phi, D$)
Generate_Binary_Tree(T,D)
for $i:\ 0 \rightarrow\ \lambda$ do
    for all $d_x \in D$ do
        $W_{\ d_x} = RandomWalk(G,\ d_x, l)$
        for all $d_y$ in $W_{\ d_x}$ do
            $J(\phi) =\ -\log Pr\left(d_k|\phi(d_y)\right)$
        end for
    end for
end for
```

D. Trust attack detection

After computing the low dimensional trust network structure using Trust2Vec, the resulting embeddings are used to detect trust attacks. We propose an algorithm which can detect large-scale bad-mouthing and self-promoting attacks, as shown in Algorithm 2. Given the network embeddings of trust graph $\phi(G)$, the source device d_s that reports the trust relationship (trustor), and the destination device d_d that received trust level (trustee). In case of a positive trust report, it is checked for possible large-scale self-promoting. Lines 1-8 checks if the similarity of embeddings vectors of the trustor and trustee is greater than the embedding similarity threshold α, the set of suspected devices is denoted as Ω_P, which is the union of the previous positive trustees and trustors of the trustor device d_s. The set of malicious self-promoting cluster M_s is determined by comparing the positive trust report to devices within Ω_P, and to devices outside Ω_P, which are denoted as $\overline{\Omega_P}$, and classified as malicious nodes if the cardinality difference is greater than the self-promoting similarity threshold β. In the case of a negative trust report, it is checked against bad-mouthing attack as shown in lines 9-20. The set of suspected device is denoted as Ω_N, which contain the devices $N_{in}(d_d)$ that previously given negative trust report against device d_d. If two or more devices within Ω_N have low dimension similarity greater than the bad-mouthing similarity threshold γ, then these nodes are classified as malicious bad-mouthing community M_b.

IV. Evaluation

A. Evaluation baselines

To test the effectiveness of the proposed system, we have compared its performance with the following trust management systems from the literature.

Algorithm 2 Trust_Attack_Detection

```
Input
Network embeddings of trust graph $\phi(G)$
$d_s$ trustor device
$d_d$ trustee device
Output
$\mathrm{M}_s$ malicious self-promoting community
$\mathrm{M}_b$ malicious bad-mouthing community

1:  if ($R\left(d_s, d_d\right) > 0$) then
2:      if (Sim($\phi\left(d_s\right), \phi(d_d)$)$> \alpha$) then
3:          $\Omega_P \leftarrow P_{in}(d_s) \cup P_{out}(d_s)$
4:          for all $d_i \in \Omega_P$ do
5:              if $\left(\left|P_{out}\left(d_i, \Omega_P\right)\right| - \left|P_{out}(d_i, \overline{\Omega_P})\right|\right) > \beta$ then
6:                  $\mathrm{M}_s \leftarrow \mathrm{M}_s \cup \{d_i\}$
7:              end if
8:          end for
9:      else
10:         $\Omega_N \leftarrow N_{in}(d_d)$
11:         for all $d_i \in \Omega_N$ do
12:             if $\left(\left|N_{out}(d_i, d_d)\right|\right)$ then
13:                 for all $d_j \in \Omega_N - \{d_i\}$ do
14:                     if $\left(Sim\left(\phi\left(d_s\right), \phi\left(d_d\right)\right) > \gamma\right)$ then
15:                         $\mathrm{M}_b \leftarrow \mathrm{M}_b \cup \{d_i, d_j\}$
16:                     end if
17:                 end for
18:             end if
19:         end for
20:     end if
21: end if
```

DDTMS [10]: In this system, the devices in the cluster start to detect the nearby devices within sensing range, and compute their trust value, and report that to a pre-elected cluster head. The latter calculates the aggregated trust score of each device in the cluster. The cluster head is periodically reelected by the network devices located within the cluster. In this system, the trust value is computed as: $Tij{=}\alpha{\times}DTij{+}\beta{\times}RTj$. Whereas, α is the weight of weight and β is the weight of indirect observation, such that α+β=1.

T-D2D [14]: In this system, the overall trust level is computed by aggregating the direct trust level that account for the direct interaction between the two devices, and indirect trust that rely on other devices' recommendations. The total trust level between device i and device j is calculated as: $TTL_{i,j} = (1-\omega)DTL_{i,j} + \omega ITL_{i,j}$, where $DTL_{i,j}$ denotes direct trust level between device i and device j, d denotes inderct trust level between device i and device j, and ω is the attention factor.

Liu-Trust: is a semi-centralized TMS that leverage blockchain for trust score management. The total trust value of device i in device j is denoted by $T_i^j(t) = \alpha{\times}DT_i^j(t){+}\beta{\times}IT_i^j(t)$, which is computed by aggregating direct trust value $DT_i^j(t)$, as well as indirect trust value $IT_i^j(t)$. The direct trust is computed as $DT_i^j\left(t\right) {=} ResT_i^j\left(t\right) \times RaT_i^j(t)$, where $ResT_i^j\left(t\right)$ is the response trust of device i to device j, and is defined as

TABLE I: Simulation parameters

Parameter	Value
OMNet++	V5.7.0
INET	V3.7.1
Mobility Type	Linear Mobility
Mobility speed	10mps
Update Interval	100ms
Transmitter power	3.5mW

the probability of device i in whether device j can provide information on time, and $RaT_i^j(t)$ is the rating trust and it represents the trustworthiness of device i about another device j on the aspect that device j can provide reliable data on the request of i.

LightTrust: is a light-weight TMS, in which the trust between two nodes p and q is estimated by aggregating: device compatibility $com_{p\to q}$, cooperativeness $coop_{p\to q}$, and delivery ratio $dl_{p\to q}$ as shown in eq (x), in addition to experience and previous knowledge.

$$T\left(p,q\right) = \sum\left(com_{p\to q}, coop_{p\to q},\ dl_{p\to q}\right)$$

DTMS-IoT [15]: is a TMS that alleviate dishonest trust recommendations and related attacks by clustering the devices using k-means algorithm.

B. Experiments

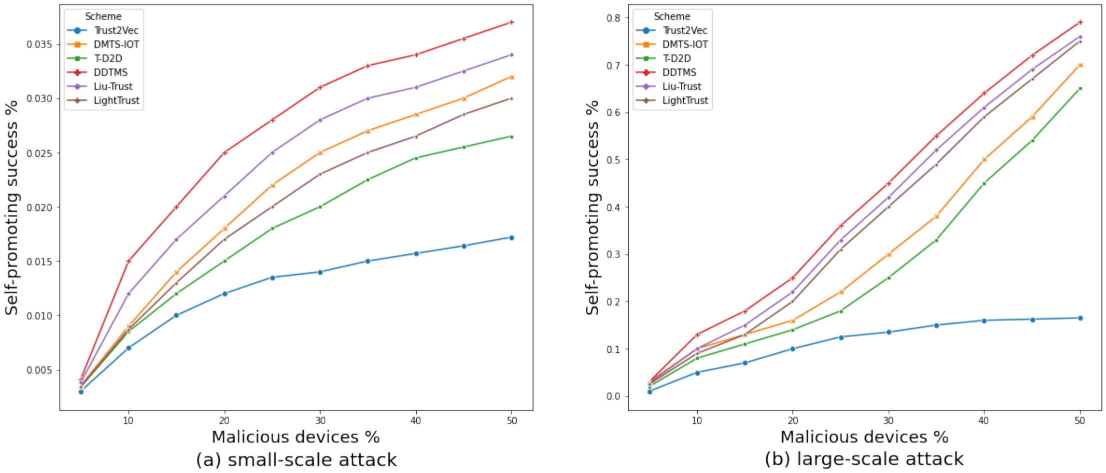
We have compared the proposed system with the above-mentioned baselines in various scenario and experiment settings. The simulation is performed using INET, an open-source model library framework of OMNet++ simulator [33], that can simulate wired and wireless networks, and also support mobility module that can be used to simulated IoT and Fog/Edge computing networks. Table I shows the simulation environment details. We simulate the network with different number of devices and malicious devices percentages to observe the effect on the overall performance. We simulate 10000 devices, The mobile devices are randomly placed at the beginning of simulation, and moves according to INET's linear mobility model. The trust scores of the devices within each cluster are stored in a local fog server that is usually the gateway to the external network. The trust value of each device is initialized as 0, and can vary between -1 and 1. Devices can express their trust level regarding their neighbouring device following device to device interaction such as data exchange interactions, or common computational or data offloading tasks. For small-scale attacks, we randomly choose a device that tries to self-promote or bad-mouth one of its neighbors that is randomly selected, we repeat the attack until the number of fake trust report submitted by the attackers represent a certain percentage (attack density) of the total submitted trust reports. We evaluate the system's performance with difference attack densities (5% to 50%). We also evaluate the system with different malicious devices percentages, in which we randomly select a certain percentage of devices to perform self-promoting or bad-mouthing attacks, we have simulated the system in different malicious devices percentage settings (5% to 50%). For large-scale attacks, we randomly select x (depending on the malicious devices percentage) devices as a group malicious devices that self-promote each others, in which each device iterate and self-promote all the other devices within the malicious group, as shown in the example in Figure 2 (a). for bad-mouthing, we randomly select a victim device, all the devices in the malicious group will bad-mouth the victim device, as shown in the example in Figure 2 (b).

We evaluate the proposed system and other baselines based on the following metrics: (1) Self-promoting attack resilience: The ability to accurately detect small-scale and large-scale self-promoting attacks without mistakenly blocking legitimate devices. (2) Bad-mouthing attack resilience: The ability to identify small-scale and large-scale bad-mouthing attacks without mistakenly blocking legitimate devices. The attack success rate is defined as the ratio of succeed attacks (e.g. self-promoting trust report) from all attempted attacks by all nodes in the network, as defined in eq (5), where s is the number of simulated devices, AS_i is the total succeed attacks by device i, and AA_i is the total attack attempts by device i.

$$ASR = \frac{\sum_{i=1}^{s} \frac{AS_i}{AA_i}}{s} \tag{5}$$

C. Results and discussion

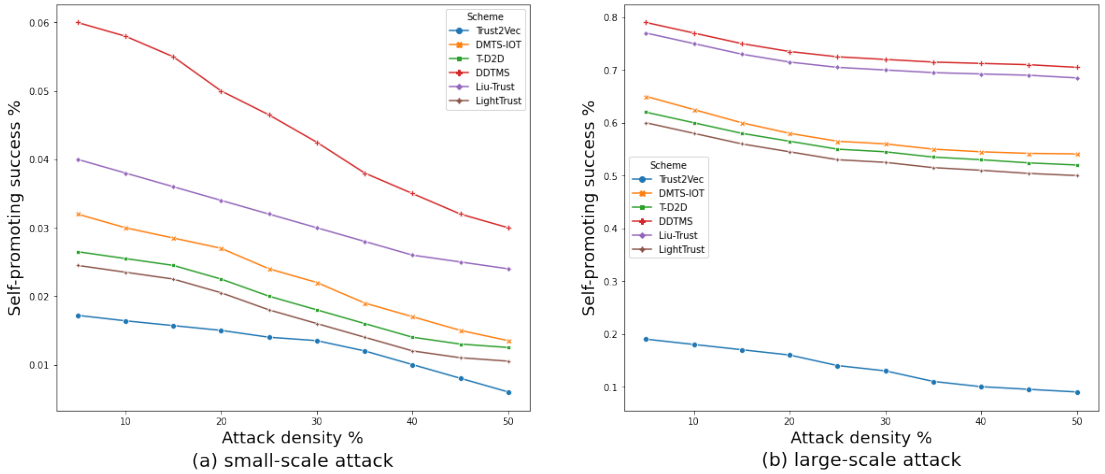
Figure 4 shows self-promoting success rate with different malicious devices percentage from 5% to 50%. In Figure 4 (a) the attack is performed as small-scale self-promotion, where the malicious devices are randomly chosen, and each two malicious devices try to inflate each other's trust value by broadcasting fake trust reports to other nodes in network. Figure 4 (b) displays a large-scale attack, where randomly selected malicious devices inflate their trust scores by distributing trust reports among their group rather than through multiple mutual trust reports, hence avoid being detected. From Figure 4(a), we can observe that the percentage of a successful self-promoting attack increases proportional with malicious devices percentage for all studied systems. All baselines have relatively low attack success rate, with 0.005% when there is 5% of the devices are malicious. The attack success rate increase when more devices malicious participate in the attack, as it become more difficult to distinguish legitimate trust ratings from self-promoting ratings; however, the attack is mitigated as the attackers are stopped once detected, and the attack success rate stabilizes with less than 0.03 for all baselines, except DDTMS and Liu-Trust, they fail to detect malicious nodes as they rely on neighboring devices' observations, which can be misleading if the neighbors are among the malicious nodes. Unlike Figure 4(a) where Trust2Vec has similar performance with the studied baselines, the upper hand of Trust2Vec is obvious in Figure 4(b) which shows the success rate of large-scale self-promoting attack with different malicious devices percentage from 5% to 50%. We can observe that Trust2Vec is the only system that can mitigate attacks as the percentage of malicious devices increases. That is because Trust2Vec analyses not only the direct trust link but also the latent trust graph structure, whereas other baselines focus on direct trust links between devices.



(a) small-scale attack

(b) large-scale attack

Fig. 4: Self-promoting in various malicious devices count



(a) small-scale attack

(b) large-scale attack

Fig. 5: Self-promoting in various attack densities

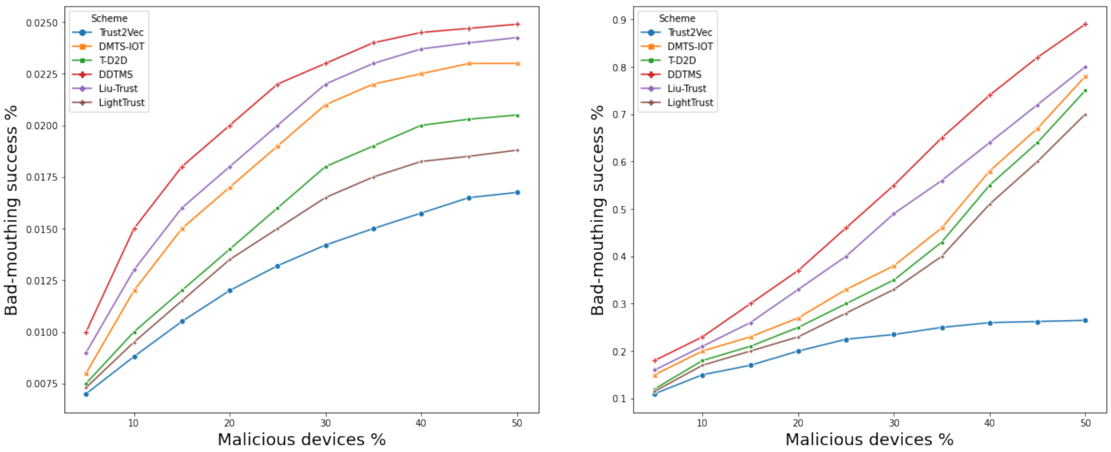
Figure 5 shows self-promoting success rate with different attack density. The higher the attack density, the more malicious devices attempt to inflate their trust score. Figure 5 (a) and (b) show small-scale and large-scale attack scenario. From Figure 5 (a), we can observe that all the attack success rate plummets when the attack density increases, that is because in high attack density, malicious device become more aggressive by sending successive fake trust reports, hence they are easily detected and eventually blocked. In Figure 5 (b), we can easily observe that Trust2Vec copes well with increase of attack density in large-scale scenarios, unlike other baselines that could not mitigate large-scale attacks.

Figure 6 shows bad-mouthing success rate with different malicious devices percentage from 5% to 50%. Figure 6 (a) and (b) display small-scale and large-scale attacks respectively. As seen in self-promoting attacks, the studied baselines have similar performance in small-scale settings. Nonetheless, the superiority of Trust2vec is obvious in large-scale settings, that is because the bad-mouthing is traced back by analysing the malicious devices' latent trust network structure through network embedding comparison, unlike the studied baselines that rely solely on the direct observation of neighboring devices.

Figure 7 shows bad-mouthing success rate with different attack density. With small-scale attack in Figure 7 (a), we can observe that all the studied baselines perform better when with higher attack density, with 0.04% attack success rate at worst (DDTMS) when the attack is at 5% density. As the attack density increases, it become much easier to detect and block malicious nodes. However, in large-scale attack scenario shown in Figure 7 (b), all baselines (DDMTS, T-D2D, Liu-Trust, LightTrust and DMTS-IOT) fail to detect malicious nodes, with at least 55% attack success rate in all attack densities. Nevertheless, Trust2Vec was able to mitigate large-scale by 82% in less dense attacks and up to 90% percent in highly dense attacks.

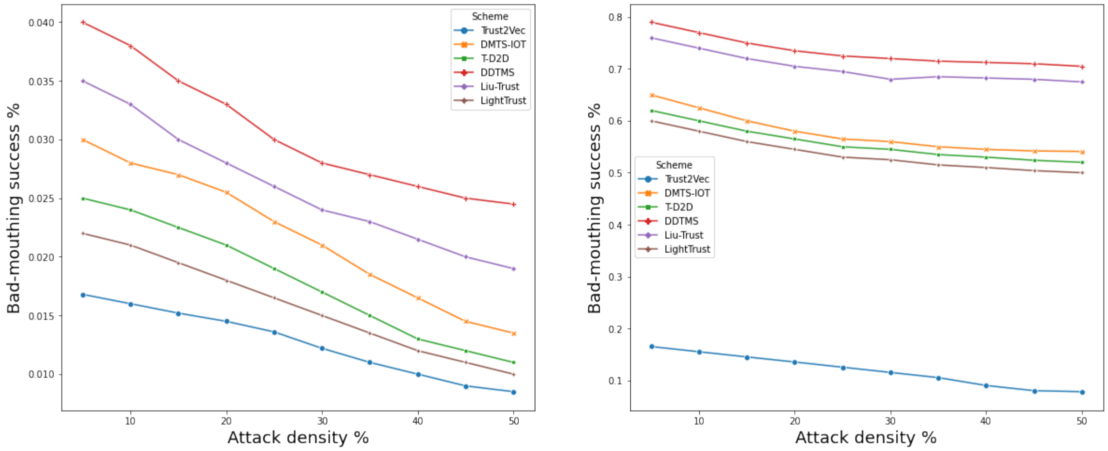
V. Conclusion

In this paper we have proposed a trust management system for large-scale IoT systems named Trust2Vec. Unlike state-of-the-art trust frameworks that focus only on small-scale IoT networks, Trust2Vec can be leveraged to manage trust relationships among devices in large-scale IoT application. Trust2Vec had been validated through large-scale IoT network



(a) small-scale attack (b) large-scale attack

Fig. 6: Bad-Mouthing in various malicious devices count



(a) small-scale attack (b) large-scale attack

Fig. 7: Bad-Mouthing in various attack densities

simulation. The results show that Trust2Vec can achieve up to 94% mitigation rate in various network scenarios. The proposed trust management system can be further improved from various aspects:

- The proposed system focus on general IoT applications, where the devices can include fixed devices such as sensors and mobile devices such as mobile phones. However, things may differ in high dynamic environments such as vehicular network. Extending the proposed system to be customized for scenario-specific IoT applications is one of our future directions
- Trust2Vec can be extended to manage trust in of virtual network entities by a software defined network.
- The proposed system manages trust scores of network devices. In our next work, we will extend that to include trust management of data entities as well.

Acknowledgment

This work was supported by:

The Insight Centre for Data Analytics funded by Science Foundation Ireland under Grant Number 12/RC/2289 P2.

CONSUS project funded by the SFI Strategic Partnerships Programme (16/SPP/3296) and is co-funded by Origin Enterprises Plc.

The National Natural Science Foundation of China under Grant 61872038.

References

[1] S. Dhelim, H. Ning, F. Farha, L. Chen, L. Atzori, and M. Daneshmand, “Iot-enabled social relationships meet artificial social intelligence,” *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 817–17 828, 2021.

[2] R. Kumar and R. Sharma, “Leveraging blockchain for ensuring trust in iot: A survey,” *Journal of King Saud University-Computer and Information Sciences*, 2021.

[3] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for internet of things,” *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.

[4] N. Kandhoul, S. K. Dhurandher, and I. Woungang, “T_cafe: a trust based security approach for opportunistic iot,” *IET Communications*, vol. 13, no. 20, pp. 3463–3471, 2019.

[5] K. Kalkan and K. Rasmussen, “Trusd: Trust framework for service discovery among iot devices,” *Computer Networks*, vol. 178, p. 107318, 2020.

[6] P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding,” *IEEE transactions on knowledge and data engineering*, vol. 31, no. 5, pp. 833–852, 2018.

[7] J. Guo, A. Liu, K. Ota, M. Dong, X. Deng, and N. Xiong, "ITCN: an intelligent trust collaboration network system in IoT," *IEEE Transactions on Network Science and Engineering*, 2021.

[8] J. Liang, W. Liu, N. N. Xiong, A. Liu, and S. Zhang, "An intelligent and trust UAV-assisted code dissemination 5G system for industrial Internet-of-Things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2877–2889, 2021.

[9] P. Kumar, G. P. Gupta, and R. Tripathi, "TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning," *Journal of Systems Architecture*, vol. 115, p. 101954, 2021.

[10] 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, 2019.

[11] R. Chen, J. Guo, D.-C. Wang, J. J. P. Tsai, H. Al-Hamadi, and I. You, "Trust-based service management for mobile cloud IoT systems," *IEEE transactions on network and service management*, vol. 16, no. 1, pp. 246–263, 2018.

[12] M. N. Ba-hutair, A. Bouguettaya, and A. G. Neiat, "Multi-perspective trust management framework for crowdsourced IoT services," *IEEE Transactions on Services Computing*, 2021.

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

[14] Z. Movahedi and Z. Hosseini, "T-D2D: A trust model for service offloading in device-to-device communication," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 10, p. e3686, 2019.

[15] O. B. Abderrahim, M. H. Elhedhili, and L. Saidane, "DTMS-IoT: A Dirichlet-based trust management system mitigating On-Off attacks and dishonest recommendations for the Internet of Things," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–8.

[16] Y. Liu, C. Zhang, Y. Yan, X. Zhou, Z. Tian, and J. Zhang, "A semi-centralized trust management model based on blockchain for data exchange in iot system," *IEEE Transactions on Services Computing*, 2022.

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

[18] M. Okuda, S. Satoh, Y. Sato, and Y. Kidawara, "Community detection using restrained random-walk similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 89–103, 2019.

[19] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "T-coin: Dynamic traffic congestion pricing system for the internet of vehicles in smart cities," *Information*, vol. 11, no. 3, p. 149, 2020.

[20] N. Aung, W. Zhang, K. Sultan, S. Dhelim, and Y. Ai, "Dynamic traffic congestion pricing and electric vehicle charging management system for the internet of vehicles in smart cities," *Digital Communications and Networks*, vol. 7, no. 4, pp. 492–504, 2021.

[21] N. Aung, S. Dhelim, L. Chen, W. Zhang, A. Lakas, and H. Ning, "Vesonet: Traffic-aware content caching for vehicular social networks based on path planning and deep reinforcement learning," *IEEE Transactions on intelligent transportation systems*, 2021.

[22] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "Accident prediction system based on hidden markov model for vehicular ad-hoc network in urban environments," *Information*, vol. 9, no. 12, 2018.

[23] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2389–2406, 2018.

[24] R.-F. Liao, H. Wen, J. Wu, F. Pan, A. Xu, H. Song, F. Xie, Y. Jiang, and M. Cao, "Security enhancement for mobile edge computing through physical layer authentication," *IEEE Access*, vol. 7, pp. 116 390–116 401, 2019.

[25] S. Dhelim, N. Aung, M. A. Bouras, H. Ning, and E. Cambria, "A survey on personality-aware recommendation systems," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2409–2454, 2022.

[26] W. Wang, H. Ning, F. Shi, S. Dhelim, W. Zhang, and L. Chen, "A survey of hybrid human-artificial intelligence for social computing," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 468–480, 2021.

[27] X. Cai, H. Ning, S. Dhelim, R. Zhou, T. Zhang, Y. Xu, and Y. Wan, "Robot and its living space: A roadmap for robot development based on the view of living space," *Digital Communications and Networks*, 2020.

[28] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, "A novel framework for mobile edge computing by optimizing task offloading," *IEEE Internet of Things Journal*, 2021.

[29] S. Dhelim, H. Ning, N. Aung, R. Huang, and J. Ma, "Personality-aware product recommendation system based on user interests mining and metapath discovery," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 86–98, 2020.

[30] S. Dhelim, H. Ning, and N. Aung, "Compath: User interest mining in heterogeneous signed social networks for internet of people," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 7024–7035, 2020.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[33] A. Varga, "Omnet++," in *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

Sahraoui Dhelim is a postdoctoral researcher at University College Dublin, Ireland. He was a visiting researcher at Ulster University, UK (2020-2021). He obtained his PhD degree in Computer Science and Technology from the University of Science and Technology Beijing, China, in 2020. And a Master's degree in Networking and Distributed Systems from the University of Laghouat, Algeria, in 2014. He serves as workshop chair of Cyberspace congress (CyberCon). His research interests include Social Computing, Digital Agriculture, Deep-learning, Recommendation Systems and Intelligent Transportation Systems.

Nyothiri Aung is a postdoctoral researcher at University College Dublin, Ireland. She received her PhD in Computer Science and Technology from University of Science and Technology Beijing, China, 2020. And a Master's of Information Technology from Mandalay Technological University, Myanmar, 2012. She worked as a tutor at the Department of Information Technology in Technological University of Meiktila, Myanmar (2008-2010). And System Analyst of ACE Data System, Myanmar (2012-2015). Her research interests include Social Computing, Medical image analysis, and Intelligent Transportation Systems.

Mohand Tahar Kechadi is a full professor in school of computer science, University College Dublin, Ireland. He received master's and Ph.D. degrees in computer science from the University of Lille 1, France. His research interests include data mining, distributed data mining heterogeneous distributed systems, grid and cloud computing, and digital forensics and cyber-crime investigations. He is a member of the Communications of the ACM journal and IEEE Computer Society. He is an Editorial Board Member of journal of Future Generation Computer Systems.

Huansheng Ning Received his B.S. degree from Anhui University in 1996 and his Ph.D. degree from Beihang University in 2001. Now, he is a professor and vice dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research focuses on the Internet of Things and general cyberspace. He is the founder and chair of the Cyberspace and Cybermatics International Science and Technology Cooperation Base. He has presided many research projects including Natural Science Foundation of China, National High Technology Research and Development Program of China (863 Project). He has published more than 150 journal/conference papers, and authored 5 books. He serves as an associate editor of IEEE Systems Journal (2013-Now), IEEE Internet of Things Journal (2014-2018), and as steering committee member of IEEE Internet of Things Journal (2016-Now).

Liming Chen is a professor in the School of Computer Science and Informatics at University of Ulster, Newtownabbey, United Kingdom. He received his B.Eng and M.Eng from Beijing Institute of Technology (BIT), Beijing, China, and his Ph.D in Artificial Intelligence from De Montfort University,UK. His research interests include data analysis,ubiquitous computing, and human-computer interaction. Liming is a Fellow of IET, a Senior Member of IEEE, a Member of the IEEE Computational Intelligence Society (IEEE CIS), a Member of the IEEE CIS Smart World Technical Committee (SWTC), and the Founding Chair of the IEEE CIS SWTC Task Force on User-centred Smart Systems (TF-UCSS). He has served as an expert assessor, panel member and evaluator for UK EPSRC (Engineering and Physical Sciences Research Council, member of the Peer Review College), ESRC (Economic and Social Science Research Council), European Commission Horizon 2020 Research Program, Danish Agency for Science and Higher Education, Denmark, Canada Foundation for Innovation (CFI), Canada, Chilean National Science and Technology Commission (CONICYT), Chile, and NWO (The Netherlands Organisation for Scientific Research), Netherlands.

Abderrahmane Lakas is a Professor at the Computer and Network Engineering department in the College of IT at UAE University. He holds an MS, and PhD in Computer Systems from the University of Pierre et Marie Curie (Paris VI, France). He has several years of both academic and industrial experience. He spent two years as a postdoc researcher at School of Computing and Communication at the University of Lancaster in UK. He is the head of CAST (Connected Intelligent Autonomous Systems) research group and the Connected Autonomous Intelligent Systems Lab (ASIL). Prior to joining UAE University, he held several industrial positions in several companies in Canada and the US including at Netrake (Plano, TX, USA), Nortel Networks (Ottawa, Canada), and Newbridge (Ottawa, Canada). His current research interests include intelligent transportation systems, vehicular ad hoc networks, unmanned ground and aerial vehicles, autonomous systems, smart cities, Internet of Things, and QoS. Dr. Lakas has published several research papers in scholarly journals. He is member of the TPC and reviewer of several renown conferences and serves in the editorial board of few journals.