Edge Intelligence for Real-Time IoT Service Trust Prediction

Prabath Abeysekara, Hai Dong, *Senior Member, IEEE* and A. K. Qin, *Senior Member, IEEE*

Abstract—Mobile Edge Computing (MEC)-based Internet of Things (IoT) systems generate trust information in a real-time and distributed manner. Predicting trustworthiness of IoT services in such an MEC environment requires new prediction strategies that cater for the aforementioned characteristics of trust information. More importantly, it is imperative to investigate how the real-time trust information could be effectively integrated into trust prediction strategies in order to capture the ever-evolving nature of trustworthiness of IoT services. In turn, such a strategy allows IoT service consumers to derive more relevant and accurate trust-based decisions. To that end, our work models trust prediction in MEC-based IoT systems as an online regularized finite-sum problem in a distributed MEC environment with a given MEC topology. We then adopt the Online Alternating Direction Method (OADM) to effectively train trust prediction models in parallel over the distributed MEC environment. OADM allows splitting the aforementioned finite-sum problem into multiple sub-problems that correspond to different local MEC environments. These sub-problems can then be solved iteratively within each local MEC environment by using the local trust data therein. This can avoid the movement of data across the core networks of mobile network providers. Experiments on real-world and synthetic datasets demonstrate the effectiveness and scalability of the proposed method.

Index Terms—Trust, Internet of Things, Mobile Edge Computing, Machine Learning, Online Learning

1 Introduction

Mobile Edge Computing (MEC)-based IoT systems are characterized by IoT services deployed in geographically distributed computing environments. These services provide useful functionalities to service consumers in close proximity [1]. Co-located within the base stations of mobile network providers, MEC environments provide computing and storage resources to applications at the edge of the network. This allows IoT services to be deployed at the edge of the network providing faster access to their consumers [2]. Such a system architecture also allows high-volume and high-velocity IoT data to be processed within MEC environments closer to where the data originates [3]. This helps reduce the network stress on the core networks of mobile network providers considerably.

Trust in such a system is an essential element. It improves the confidence of service consumers towards achieving the desired outcomes when interacting with IoT services. For instance, Fitness tracking applications collect personal information from users such as social profile, behavioral and location data via wearable devices and other means. This information is, then, shared with third party services. In such a context, trust provides *assurance* that the collected information will be used *as agreed* by all parties [4]. In intelligent transport systems (ITS), trust allows autonomous vehicles to locate services that provide credible location and traffic information [5]. As such, trust can be deemed an integral part towards ensuring user acceptance towards consuming services in IoT systems.

However, *the dynamic and heterogeneous nature in MEC-based IoT environments poses multiple challenges in the context of determining the trustworthiness of IoT services*.

1) Mobilizing IoT sensors and service consumers cause the trustworthiness of IoT services to be constantly *re-evaluated*: Let us take an example of autonomous vehicles acting as navigation data providers for a navigation information service in an ITS. An autonomous vehicle entering and leaving the coverage area of a given MEC environment can influence the QoS of the navigation services to vary sporadically *in real-time*. In addition, the QoEs of one service consumer towards a given IoT service may be different from that of another, as well. As a result, the trustworthiness of an IoT service as perceived by one service consumer may be different from the other service consumers operating within a given MEC environment. Therefore, the trust information generated within a given MEC environment from the transactions amongst mobilizing IoT sensors (e.g. navigation data providers) and consumers can cause the trust dynamics of IoT services observed within a MEC environment to change in real-time. *This requires the trust evaluation models to be continuously updated*.

2) *Real-time* trust information generated in distributed MEC environments under *heterogeneous* operating conditions demand *real-time, context-aware* and *distributed* trust evaluation of IoT services: In a typical MEC-based IoT system, trust information is generated in a *distributed* manner due to its inherently distributed system architecture [6]. In addition, such trust information is also generated under different operating conditions from one MEC environment to another. These operating conditions include network conditions (e.g. network congestion), computing

- *P. Abeysekara, and H. Dong are with School of Computing Technologies, RMIT University, Melbourne, VIC, Australia*
  E-mail: prabath.abeysekara@rmit.edu.au; hai.dong@rmit.edu.au
- *A.K. Qin is with Department of Computing Technologies, Swinburne University of Technology, Hawthorn, VIC, Australia*
  E-mail: kqin@swin.edu.au

Manuscript received February 28, 2022; revised, 2022; accepted, 2022. (Corresponding author: Hai Dong)

and storage resource availability, the hardware and software used to host IoT services amongst others [7]. We refer to the aforementioned operating conditions as the *context for trust*, which can vary from one MEC environment (i.e. *context-environment*) to another [8]. This gives rise to *multiple context environments* within a given MEC topology that are prone to change over time in response to fluctuations in the underlying operating conditions, as well. This demands trustworthiness of *homogeneous* IoT services within different MEC environments to be determined *in a context-aware manner subject to time-varying trust contexts*.

3) Lack of trust information with sufficient diversity IoT services may hinder the ability of MEC-local *real-time* trust evaluation strategies to produce accurate trust decisions: Due to the distributed system architecture of MEC an individual MEC environment only sees *a split view* of the entire IoT service ecosystem that exists across a given MEC topology. Therefore, a trust evaluation model trained within a given MEC environment may suffer from the lack of trust information on some IoT services (i.e. trust information *sparsity*) and lack of *diversity* in trust information (i.e. trust information collected from a wider variety of *homogeneous* IoT services). Consequently, such issues can cause an MEC-local trust prediction model to lack generalizability, which we refer to as the *ability to perform well on unseen trust prediction requests received from service consumers*. This challenges accurately determining the trustworthiness of particularly the less popular IoT services deployed in a given MEC environment.

To address the aforementioned challenges, we propose an *edge intelligence strategy to predict real-time IoT service trust within MEC-based IoT environments*. The specific contributions that address the challenges outlined previously are summarized below.

1) We formulate the trust prediction problem in an MEC-based IoT system as *an online learning problem subject to concept-drift over a set of distributed, time-varying and context-dependent trust information distributions*. *we define concept-drift as the change in trustworthiness of IoT services over time in response to a change in the context-environment under which they operate*.

2) We also propose a parallel algorithm to train a context-aware and real-time trust prediction model in a collaborative manner within an MEC topology. The aforementioned algorithm uses *Online Alternating Direction Method* (OADM) to address **challenge 1, 2** and **3**. More specifically, the proposed algorithm

- enables integrating temporally-ordered streams of real-time trust information into MEC-local trust prediction models to address **challenge 1**, thereby allowing IoT service trust to evolve continuously.
- allows training a set of distributed and context-aware MEC-local trust prediction models atop data accumulated within each individual MEC environment under heterogeneous operating environments that best meet the trust characteristics in different MEC environments, to address **challenge 2**. This also helps significantly reduce the movement of high-volume trust information across the core networks of mobile networks thereby complimenting the goals of the MEC paradigm.
- enables sharing knowledge amongst similar context-environments for trust prediction collaboratively, to address **challenge 3**. This tackles the issue of trust information *sparsity* and helps improve the accuracy of MEC-local trust prediction models in response to trust queries from service consumers.

3) We report results of our exhaustive evaluation of the proposed approach carried out against the state-of-the-art distributed and centralized trust prediction approaches in the current literature.

The rest of the paper is structured as follows: Section 2 reviews the prior research our work builds on. Section 3 describes a motivation scenario. Section 4 formally defines the problem setting. Section 5 elaborates the proposed solution. Section 6 comprehensively details out the experiments and evaluation of the proposed solution. Section 7 concludes our work and discusses possible future work.

2 Related Work

This section primarily evaluates the existing literature that precedes the proposed work across two key themes, in the form of 1) Trust evolution in IoT systems and 2) Edge Intelligence for trust prediction. We outline the key limitations in these existing works in Section 2.1 and Section 2.2, respectively.

2.1 Trust evolution in IoT systems

Trust evolution remains a relatively understudied aspect of trust in IoT systems. Among the limited number of studies that do focus on trust evolution, one noteworthy classification proposes trust update approaches to be of two folds. They can either be 1) event driven, or 2) time-driven [9]. Event-driven trust updates are often driven in response to important events triggered by transactions, changes to the trust context, topological changes of the underlying trust networks. On the other hand, time-driven trust updates often take into account the intrinsic behaviour of trust, which results in gradual decays of its value over time. [10] investigated temporal evolution of trust. This work proposed a time-aware trust prediction model based on supervised learning approach to predict evolution of trust, which uses an exponential weighting approach to give more prominence to recent observations. Meanwhile, [11] categorizes trust refreshment patterns into five main groups, namely 1) stable, 2) abrupt, 3) incremental, 4) gradual, and 5) recurring. This categorization is primarily influenced by the concept of drift, which aims to address changes in relationships among entities over time.

In addition, [12] proposed a high-level trust management framework to address the dynamicity in IoT systems caused by new devices and services entering and leaving the system unpredictably. While this approach also attempts to model context-awareness in trust, the aforementioned context is predominantly defined over an individual user. [13] proposed another approach to detect dynamicity of trustworthiness of IoT services. This approach, however, is only applicable for centralized IoT systems as it does not

account for the first principles of trust information generation and processing in an MEC-based IoT environment. Meanwhile, a dynamic trust model for collaborative applications in IoT systems was proposed in [14]. [15] proposed an adaptive trust evaluation model for crowd-sourced IoT services. This approach considers the dynamic changes in the trustworthiness of crowd-sourced IoT services based on consumers' usage of them. This aligns with the fundamentals of evolving trustworthiness of IoT services proposed in our work. Apart from that, a framework is proposed for crowdsourcing services to IoT devices based on their mobility and trustworthiness [16]. [17], meanwhile, proposed a dynamic trust management protocol to assess the trustworthiness of misbehaving nodes based on honesty, cooperativeness and community of interest. This approach is capable of adjusting the trustworthiness of underlying nodes based on the changing environmental conditions and requires no centralized trust authority. Furthermore, two conceptual frameworks to address the problem of trust evaluation in dynamic IoT systems were proposed in [18] and [16]. The aforementioned two approaches only address the underlying problem at a high-level solution level, and do not recommend concrete implementations to achieve their proclaimed goals. In addition, they both have been proposed predominantly in the context of centralized IoT systems, and therefore, can be deemed less suitable to a decentralized setting as that of an MEC-based IoT system.

2.2 Edge intelligence for trust prediction

As edge-oriented IoT systems have been taking traction in the recent past, there has been an increasing interest in using edge intelligence strategies for trust evaluation. [1] proposed a distributed and collaborative edge intelligence strategy for IoT service trust prediction that runs in *batch-mode* atop historical trust information accumulated in each MEC environment within a given MEC topology. [6] proposed a data-driven, context-aware and stochastic edge intelligence approach for trust prediction in MEC-based IoT systems. This approach aimed to tackle the challenges posed by the varying trust characteristics and high-volume trust information accumulated within different MEC environments in growing IoT networks. Meanwhile, [19] proposed a data-driven edge intelligence strategy for network anomaly detection, whereas [20] proposed an approach to evaluate trustworthiness of edge nodes in order to improve security and privacy of edge-based IoT systems. However, none of these approaches consider the evolutionary aspect of trustworthiness nor facilitate real-time prediction of IoT service trust. In addition, these approaches also fail to allow efficient trust prediction in the face of mobilizing IoT services and consumers as well as address the sparsity of trust information that can cause trust inferences to be less relevant. [21] acknowledges the dynamicity and context-dependence of trust within a collaborative edge computing environment in relation to content cache placement. However, it does not take into account the key requirements and characteristics of IoT services and their consumers outlined before. Furthermore, [22] introduced a multi-criteria DoS attack detection approach for MEC-based IoT systems. Not only does it fail to counter the effect of trust information sparsity, but it also relies heavily on network-level parameters associated with IoT devices for trust modelling, which might not always be available to detect the trustworthiness of IoT services deployed in a service oriented IoT environment.

3 Motivating Scenario

We use the following motivating scenario to highlight the relevance of this work in relation to the challenges outlined in Section 1.

Let us take an example of an autonomous vehicle, AV_1, operating in an MEC-based IoT environment (see Fig. 1). AV_1 needs to obtain real-time navigation information from *trustworthy* navigation information services to take its passengers from the origin of the trip, X to the desired destination, Y. This navigation information could be provided by automotive vendors themselves or third-parties within this MEC-based IoT environment[1]. For simplicity, we assume that these navigation services offer similar key functionalities to their consumers, and therefore, form a *functionally homogeneous group of services*. Meanwhile, the navigation information provided by these services could be supplied by other autonomous vehicles themselves, non-autonomous vehicles with sensing equipment planted on them via crowd-sourcing[2,3], Unmanned Aerial Vehicles, or stationary stationary roadway sensors and traffic surveillance systems [23]. Each MEC environment provides a platform for this data to be made available *as services* to AV_1 and other interested consumers [24]. As AV_1 goes past multiple adjacent MEC environments $M_{(AV_1;X,Y)} = \{M_1, \ldots, M_k\}$ enroute to point Y from X, it interact with trustworthy navigation information services that matches its QoEs. We assume that these QoEs of AV_1 would remain the same as it goes past MEC environments in $M_{(AV_1;X,Y)}$.

As AV_1 consumes the *homogeneous* navigation information services exposed by $M_{(AV_1;X,Y)}$, it will generate information that characterizes the trustworthiness of these services. This information can potentially include Quality of Expectations (QoEs; e.g. expected service latency, freshness and accuracy of the navigation information provided.) of AV_1, Quality of Service (QoS) values of the service at the time of consumption, environmental conditions under which the service was operating, as well as *user satisfaction levels* or *ratings* that determine if the interaction yielded a positive or negative outcome for AV_1. We propose this trust information could take the shape of the tuple $\langle t, QoE_{(AV_1,t)}, QoS_{(S_{M_j},t)}, OC_{(M_j,t)}, Res\rangle$ where t denotes the time of the transaction, $QoE_{(AV_1,t)}$ denotes the QoEs of AV_1 at time t, $QoS_{(S_{M_j},t)}$ denotes the QoS of the service S_1 deployed in the MEC environment M_j consumed by AV_1 at time t, $OC_{(M_j,t)}$ denotes the operating conditions of the MEC environment M_j at time t and Res denotes the outcome of the interaction between AV_1 and S_1, as perceived by AV_1.

In this problem setting, the mobilizing autonomous vehicles acting as navigation information providers can

[1]https://www.wired.com/story/your-next-gig-map-the-streets-for-self-driving-cars/

[2]https://mapper.ai/

[3]https://www.esri.com/about/newsroom/publications/wherenext/crowdsourc location-intelligence-autonomous-vehicles/

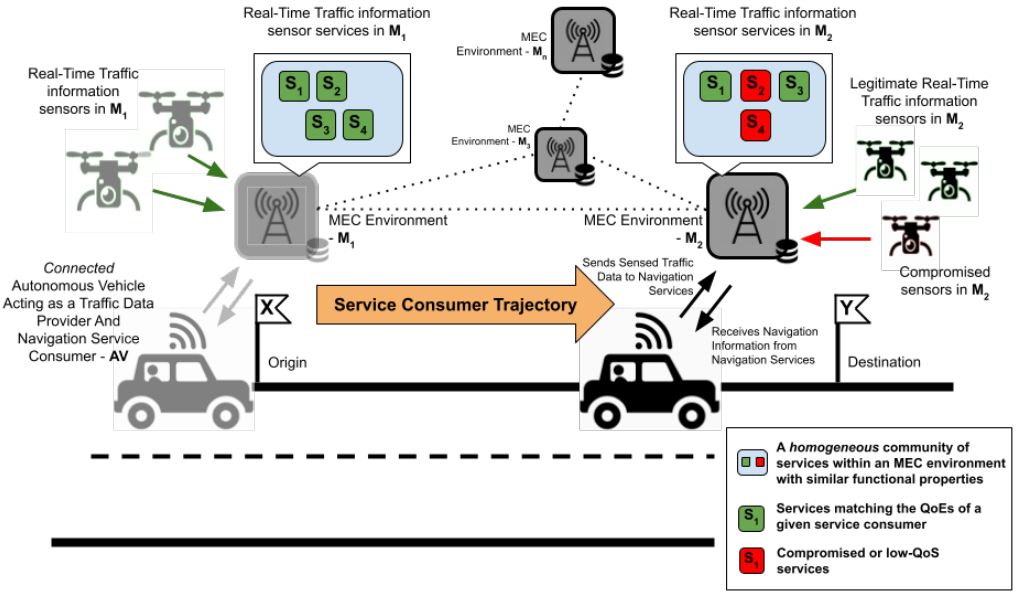

Fig. 1: A hypothetical deployment of a MEC-based traffic sensing system.

sporadically provision navigation sensor services within a new context-environment for trust. This can cause the underlying context-environment to change and the trust prediction models established within $M_{(AV_1;X,Y)}$ to be re-evaluated frequently. In addition, by nature, the values of some of the parameters that the IoT service trust depends on (e.g. QoS levels of a given service and environmental conditions) can vary over time. This phenomenon causes the trustworthiness of the underlying service to *evolve* over time. Such dynamism challenges most existing trust prediction approaches that predominantly rely on historical trust information. Furthermore, the lack of suitable and sufficiently diverse trust information when a service is provisioned in a new context-environment also prevents existing trust evaluation approaches from being effective. The aforementioned challenges demand alternative trust prediction strategies that can withstand the dynamicity of trust and adhere to the systems characteristics of MEC-based IoT systems.

4 Problem Formulation

The trust between IoT services and their consumers is bound to evolve over time as they interact with each other [17]. In other words, a positive and negative outcome of the interaction between the IoT services and their consumers will either strengthen or weaken the trust of the service consumers towards IoT services, respectively. Therefore, a real-time trust estimation strategy for IoT services has to update itself based on the temporal order in which new trust information becomes available. Accordingly, given a sequentially arriving *unbounded* stream of *real-time* trust information generated at a set of arbitrary time intervals $T = \{1, \ldots, t\}$ (where $t \rightarrow \infty$), we extend the aforementioned principle to our problem setting and formulate the problem of predicting the trustworthiness of an IoT service under temporal evolution of trust in real-time, as below.

$$\hat{y} = f^t(x; w) \tag{1}$$

where $x \in R^d$ (d represents the dimensionality of x) denotes a requirement specification (e.g. expected QoS levels) of an arbitrary service consumer looking to determine the trustworthiness of a given service organized into a vectorized form, $w \in R^d$ denotes a set of coefficients indicating the impact of each trust feature towards the overall trust value, and f^t is a *time-varying* mapping function (a.k.a *trust prediction model*) defining how each trust feature and their respective weight coefficients can be consolidated to come up with an overall trust value $\hat{y}$ at time t. Here, f^t represents the *best estimator in hindsight* for IoT service trust at time t within the underlying MEC environment, and is bound to change as more and more new trust information is made available at time intervals $\{t, \ldots, \infty\}$.

A *learner* responsible for training a MEC-local trust prediction model in a real-timely evolving trust system, at time t, only has access to trust information generated *in the past* at intervals $\{1, \ldots, (t-1)\}$. Therefore, we formulate the problem of estimating f^t at time t, as finding the best estimator of w that minimizes the *cumulative* loss denoted by ℓ^t over all past time intervals $\{1, \ldots, (t-1)\}$, as below.

$$\underset{w \in R^d}{\text{minimize}} \quad \sum_{i=1}^{(t-1)} \ell^i(\langle X^i, y^i \rangle; w) \tag{2}$$

where ℓ^i denotes the loss function revealed by the nature at time interval i, $\langle X^i, y^i \rangle$ denotes a batch of data arrived in the given MEC environment at the i^{th} time interval. In the aforementioned formulation, the learner engages in the process of training the underlying trust prediction model iteratively, as depicted in Algorithm 1 below.

As elaborated in *challenge 1 and 2* in Section 1, it is imperative that a trust evaluation system deployed within an MEC-based IoT system consistently exhibits and is representative of the most recent characteristics of underlying IoT services. To realize this goal, we intend to modify the problem (3) to treat the most recent trust information

Algorithm 1 Learner behaviour

1: **for** $i = 1, 2, \ldots, t$ **do**
2: Learner receives trust information X^i generated at time i
3: Learner estimates w_i
4: Nature provides the loss function ℓ^i and y.
5: Learner incurs the loss $\ell^i(\langle X^i, y^i \rangle; w)$ and updates the underlying trust model

collected from the transactions amongst the IoT services and their consumers to be more influential towards determining a trust prediction model for a given MEC environment, as below.

$$\underset{w \in R^d}{\text{minimize}} \quad \sum_{i=1}^{(t-1)} h(t).\ell^i(\langle X^i, y^i \rangle; w) \tag{3}$$

where $h(t) = h_0.e^{-\lambda t}$ represents an *exponential decay function* used to weigh the training examples gathered at the t_{th} time interval with respect to the other training examples gathered during the time intervals $\{1, \ldots, (t-1)\}$. In $h(t)$, h_0 represents the initial weight at time $t = 0$, and $\lambda(> 0)$ is the decay constant.

In a typical MEC-based IoT system, the real-time trust information generated from transactions between IoT services and their consumers is persisted and processed in an entirely *distributed* manner within local MEC-based data-centres. Given the heterogeneous operating conditions available in different MEC environments, we propose that the attributes in the trust information defining IoT service trust (i.e. *trust features*) x_i, or the characteristics of the trust information distributions in each MEC environment M_i could change from one MEC environment to another (i.e. *non-IIDness*). This leads to multiple *context environments* for trust prediction within different MEC environments. Moreover, we also propose that the aforementioned trust features could be formulated as a combination of parameters that are common across every MEC environment x_{co} (e.g. QoS parameters such as latency, availability, reliability) as well as those that are specific to each individual MEC environment $\bar{x}_i$. However, for simplicity, we assume $x_i = x_{co}$, with x_i representing *non-IID* trust information across different MEC environments. We also assume that the context environments originate only from different characteristics of trust information distributions caused by the varying operating conditions available in different MEC environments. Given the above, we formulate the problem of training a *distributed* and *context-dependent real-time* trust prediction model over a set of *context environments*, as below.

$$\begin{aligned} \underset{[w_1, w_2, \ldots, w_m] \in R^d}{\text{minimize}} \quad & \sum_{m=1}^{M} \left(\sum_{i=1}^{T} \sum_{j=1}^{n_i^t} h(t).\ell_i(\langle x_i^j, y_i^j \rangle; w_i) \right) \\ & = \sum_{i=1}^{m} \left(\sum_{t=1}^{T} \mathcal{L}_i(w_i) \right) \end{aligned} \tag{4}$$

where MEC environments within a given MEC topology are indexed with $m \in (1, \ldots, M)$; ℓ_i and $\mathcal{L}_i$ denote the loss and cost functions used to train a trust model in i^{th} MEC environment within a given MEC topology, w_i denotes the weight vector associated with F_i and ℓ_i, $P_i^t = \{x_i^j, y_i^j\}_{j=1}^{n_i^t}$ denotes the dataset generated and used at time t to train the trust model used by the i^{th} MEC environment M_i. Problem (4) easily parallelizable across different MEC environments by letting each MEC environment train individual real-time trust prediction models over the MEC-local trust information.

Furthermore, in a typical MEC-based IoT system, sensor data providers and consumers can mobilize among adjacent MEC environments. Not only that, it is also possible that same sensor provider or those that exhibit similar characteristics can operate from MEC-environments in close proximity to each other. As a result, we hypothesize that adjacent or MEC environments that are close to each other may accumulate overlapping trust information, and form *similar context environments* for real-time trust prediction. Therefore, allowing such MEC environments to collaborate with each other may assist MEC environments with *similar context environments* to counter the impact of *sparse* trust information, thereby allowing them to derive more *accurate* trust prediction models by *borrowing strength from each other*. Consequently, problem (3) can be further modified to incentivise knowledge sharing amongst the neighbouring MEC environments that carry similar trust features, as below.

$$\begin{aligned} w = \underset{w \in R^d}{\text{minimize}} \; & \mathcal{L}(w) + \gamma \mathcal{G}(w, \{w_i\}_{w \neq w_i}) \\ & \text{s.t.} \\ & \mathcal{G} = \sum_{i \in N(i)} \|w - w_i\|_2 \end{aligned} \tag{5}$$

where $\mathcal{G}$ infuses the knowledge (i.e. model parameters) extracted from the neighbours, γ scales the impact of knowledge acquisition against the knowledge derived from MEC-local trust information and $N(i)$ denotes the neighbouring MEC environments taking part in knowledge sharing. $\mathcal{G}$ encourages the parameters w of a trust prediction model within an MEC environment to be selected from the knowledge acquired from its neighbours either by adopting their entire model or an aggregated form of (e.g. mean) the model parameters of the neighbours, under different circumstances.

However, $\mathcal{G}$ spoils the parallelism enjoyed by (4) as it now force the trust prediction model trained with in the MEC environment M_i to depend on the model parameters of its neighbours, which need to be determined at the same time or before that of M_i. Therefore, we look to aggregate all sub-problems denoted by problem (5) that are to be solved by each MEC environment together as below, and attempt to derive a parallelizable solution.

$$\begin{aligned} & [w_1, w_2, \ldots, w_m] = \\ & \underset{[w_1, w_2, \ldots, w_m] \in R^d}{\text{minimize}} \sum_{i=1}^{\infty} \left(\sum_{m=1}^{M} \mathcal{L}_i(w_i) + \sum_{m=1}^{M} \gamma_i \mathcal{G}_i(w_m, \right. \\ & \left. \{w_j\}_{w_i \neq w_j}) \right) \end{aligned} \tag{6}$$

5 Proposed Solution

This section provides a comprehensive overview of the proposed solution and the theoretical foundation upon which

it is developed. Section 5.1 lays out an easy-to-comprehend summary of the proposed solution void of rigorous mathematical notations. Section 5.2 comprehensively details our proposed solution to address the key challenges in trust prediction outlined in Section 1 together with a brief summary of the important precursors that it relies upon.

5.1 Solution Overview

We propose a parallel and iterative distributed online predictive algorithm for trust prediction in MEC-based IoT systems. From a communication perspective, the proposed algorithm is designed to run atop a two-tier hierarchical communication architecture. This hierarchical communication architecture predominantly consists of the global cloud and MEC environments as its tiers. Furthermore, the information flows between the two aforementioned tiers via the backhaul links connecting the global cloud layer and the network layer of each distributed MEC environment. The proposed algorithm runs in six key steps, which are elaborated below.

Step 1: First, the streaming trust information generated from the transactions amongst IoT services and their consumers would be accumulated in real-time within MEC environments.This real-time trust information will be persisted in suitable time-series data storesor streaming analytics platforms to be used by the proposed iterative online trust prediction strategy (see Fig. 2(a)).

Step 2: Then, simultaneously, a Global Model Coordinator (GMC) running in the centralized cloud layer will bootstrap each MEC environment with necessary metadata to train trust prediction models. This metadata includes, 1) details about the neighboring MEC environments, 2) auxiliary model parameters of the trust prediction models trained by the neighbors to enable knowledge-sharing. In case an MEC environment is about to run its first iteration of the proposed iterative online trust prediction strategy, it will be initialized with suitable defaults of the corresponding metadata (see Fig. 2(b)).

Step 3: Next, at each iteration of the proposed iterative online trust prediction strategy, a family of online trust prediction models will be derived. This is carried out based on either a single training sample of trust information or the entire batch (given the computing resource available) of trust information accumulated within a given timeslice (see Fig. 2(c)).

Step 4: Once trained, the model parameters of the trained MEC-local online trust prediction models are, then, shared with the GMC running in the centralized cloud layer (see Fig. 2(d)).

Step 5: Now, the GMC accumulates all the model parameters it has received from all the distributed MEC environments. It then enforces knowledge sharing among neighboring MEC environments and relevant other metadata is computed (see Fig. 2(e)).

Step 6: Once knowledge sharing and the computation of the metadata is done, the GMC next shares the learnt knowledge and metadata associated with each neighboring MEC environments with every MEC environment (see Fig. 2(b)).

Step 7: The procedure formed by steps 2) to 6) are then repeated iteratively at each timeslice of the associated time horizon, which marks the end of the learning process and can be either finite (i.e. in case the trust prediction model training stops after a finite number of iterations) or infinite in case the learning process runs indefinitely.

5.2 Our Solution

This section is structured into two sub-sections. For completeness, in Section 5.2.1, we first provide a brief systematic exposition below on OADM, which is used to parallelize the key problem formulation we derived in Section 4. We then comprehensively describe the proposed solution in Section 5.2.2.

5.2.1 Online Alternating Direction Method (OADM)

OADM is an online algorithm that promotes solving a linearly-constrained optimization problem by attempting only a single pass over a given arbitrary dataset [25]. The speciality of OADM over its predecessor ADMM stems from the former requiring only a single pass over a given set of training examples, while ADMM attempts to do multiple passes over the same set of training examples across multiple iterations till it the algorithm eventually reaches convergence. Therefore, OADM could be deemed a better fit to solve a linearly constrained optimization problem in the context of data-stream learning, which demands training a prediction model incrementally over a dataset accumulated in real-time.

OADM algorithm primarily intends to take on the problems of the following type.

$$\underset{x\in X, z\in Z}{\text{minimize}} \quad \sum_{t=1}^{T}\Big(f^t(w)+g(z)\Big) \quad \text{s.t.} \quad Aw+Bz=c \quad (7)$$

where $w \in R^n, z \in R^m, A \in R^{p\times n}, B \in R^{p\times m}$. It is assumed that the functions denoted by $f(w)$ and $g(z)$ are convex and defined as $f^t : R^n$ and $g : R^m$ and is a time-varying loss function [26]. In most convex optimization problems where OADM can naturally be applied, $f(w)$ corresponds to a loss function where as $g(z)$ corresponds to a regularization function that helps better generalize the solution of the optimization problem being solved.

To solve the constrained optimization problem (7) as an unconstrained problem, the augmented lagrangian associated with it $L_p^t(w, z, \mu)$ is obtained at time t [27]. Then, by applying dual-ascent iteratively, OADM minimizes the augmented Lagrangian $L_p^t(w, z, \mu)$ with the following steps at time t.

$$w^{t+1} = \underset{w\in R^n}{\operatorname{argmin}} L_p^t(w, z^t, \mu_t) \quad (8a)$$

$$z^{t+1} = \underset{z\in R^m}{\operatorname{argmin}} L_p^t(w^{t+1}, z, \mu_t) \quad (8b)$$

$$\mu^{t+1} = \mu^t + \rho\nabla\mu L_p^t(w^{t+1}, z^{t+1}, \mu) \quad (8c)$$

When $f^t(w)$ and $g(z)$ are separable into multiple sub-problems each solved over a partition of the training data population, the aforesaid time-based iterations can be carried out to solve each sub-problem independently in parallel. The solution proposed in this work utilizes this exact behaviour to solve optimization problems on potentially large graphs as a distributed online optimization problem.

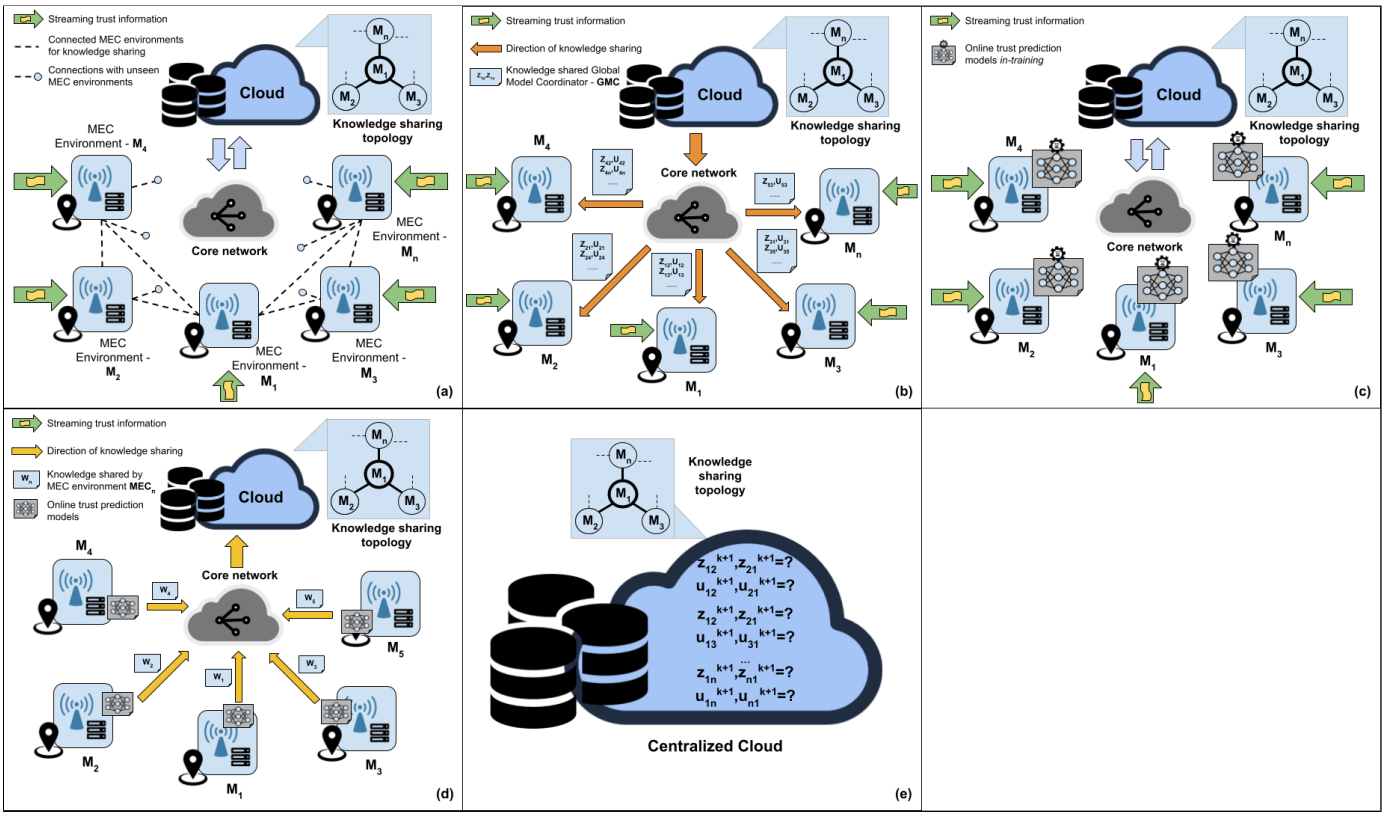

Fig. 2: A visualization of the information flow associated with the proposed solution.

5.2.2 *OADM to derive a parallel solution*

In this subsection, we present the proposed parallel solution to derive a mobility- and context-aware real-time trust prediction model for MEC-based IoT services.

In a typical MEC topology, each individual MEC environment accumulates real-time and continuous streams of trust information originated from the transactions among IoT services and consumers within geographically distributed MEC-local data centers. Furthermore, these individual MEC environments also tend to operate independently from others within their own network boundaries [28]. This can hinder their ability to share knowledge with each other. In addition, although direct communication amongst the MEC environment for knowledge sharing is possible [24], complexities in inter-MEC network communication coupled with lack of interoperability standards encouraged us to utilize the centralized cloud to facilitate knowledge sharing. Even though the MEC paradigm attempts to overcome scalability challenges posed by centralized cloud-based infrastructure in the face of high-volume IoT data, edge-cloud collaboration has attracted much attention in order to simplify communication among MEC environments [29]. In such a setting, each MEC environment can be *logically* connected to multiple MEC environments via the centralized cloud for collaborative training of trust prediction models. The trust prediction problem in MEC-based IoT systems formulated in Section 4 was then modelled over the graph resulting from this topology and OADM was applied to derive a parallel solution to train a distributed trust prediction model giving rise to Algorithm 2.

Algorithm 2 runs in multiple key steps in harmony with the cloud and MEC layers. First, the model parameters of trust prediction models trained by each MEC environment are initialized by a GMC running in the cloud layer (see lines [2-3]). After that the OADM procedure runs its three key steps denoted by problems (8a), (8b) and (8c) alternatingly between the cloud and MEC layers, as below.

w_i**-update**: Separable across each local MEC environment, w_i-update is solved iteratively in parallel atop MEC-local trust information (see Fig. 2(c)). Utilizing the z_{ij}- and u_{ij}-updates from the previous iterations shared by the GMC during the initialization phase (see Fig. 2(b)), each local MEC layer then independently trains its own local trust prediction model (see lines 7, [14-17]). Once done, all MEC environments share their resulting model parameters w_i as well as the loss incurred on using the trust prediction model trained at time horizon t atop the data accumulated at time horizon $t+1$, with the GMC (see line 17 and Fig. 2(d)).

z_{ij}**-,** z_{ji}**- and** u_{ij}**-updates:** In contrast to w_i-update, z_{ij}-, z_{ji}- and u_{ij}-updates are carried out within the cloud layer. Out of the aforementioned steps, z_{ij}-, z_{ji}- perform knowledge sharing by forcing the model parameters of the trust prediction model trained by a given MEC environment to be similar to the mean of the cluster it belongs to (see lines 9, [18-20]), while u_{ij}-updates concerns with updating dual variables used by the OADM framework (see lines 10, [21-23] and Fig. 2(e)).

Output: The *evolving* output produced by the OADM procedure (see line 13) at each time horizon t consists of the model parameters of each individual MEC environment corresponding to their trust prediction models.

We used a soft-margin Support Vector Machine (SVM)

Algorithm 2 Mobility- and context-aware real-time trust prediction of MEC-based IoT services

parameters: M-MEC environments, E-Connectivity among MEC environments for knowledge sharing, ρ-Penalty parameter, $\mu(=10), \nu(=2)$-Residual balancing parameters, k-Sliding window length for model averaging.
for all $m \in M$ **do** ▷ Loop over MECs in Cloud layer
  Send initial z_{ij}, z_{ji} and u_{ij} to m
procedure OADM($w_i^t, z_{ij}^t, u_{ij}^t$)
  for $t = 1$ to T **do**
    for all $m \in M$ **do** ▷ Loop over MECs in parallel
      $w_i^{t+1}, C_{f_i}^{t+1} \leftarrow$ W-UPDATE($w_i^t, z_{ij}^t, u_{ij}^t$)
    for all $e \in E$ **do** ▷ Loop over edges among MECs, in cloud layer
      $z_{ij}^{t+1}, z_{ji}^{t+1} \leftarrow$ Z-UPDATE(w_i^{t+1}, u_{ij}^t)
      $u_{ij}^{t+1} \leftarrow$ U-UPDATE(w_i^{t+1}, z_{ij}^{t+1})
      Compute loss $C_{ed_{ij}}^{t+1} \leftarrow \|z_{ij} - z_{ji}\|_2$
    Compute loss $C_{tot}^{t+1} \leftarrow \sum_{m \in M} C_{f_i}^{t+1} + \sum_{(j,k) \in E} C_{ed_{jk}}^{t+1}$
    Compute total constraint violations - r^t
    Compute dual residual - s^t
    $\rho \leftarrow$ RHO-UPDATE(r^t, s^t, ρ)
    return $\frac{1}{k} \sum_{(t-k)}^{t} W$
procedure W-UPDATE($w_i^t, z_{ij}^t, u_{ij}^t$)
  $w_i^{t+1} \leftarrow \underset{w_i}{\operatorname{argmin}} \Big(f_i^t(w_i) + \sum_{j \in N(i)} \frac{\rho}{2} \|w_i - z_{ij}^t + u_{ij}^t\|_2^2 + \frac{1}{2\sqrt{t}} \|(w_i - w_i^t\|_2^2 \Big)$
  Compute loss $C_{f_i}^{t+1} = f_i^{t+1}(w_i^{t+1})$
  Send $w_i^{t+1}, C_{f_i}^{t+1}$ to cloud layer
procedure Z-UPDATE(w_i^{t+1}, u_{ij}^t)
  $z_{ij}^{t+1}, z_{ji}^{t+1} \leftarrow \underset{z_{ij}, z_{ji}}{\operatorname{argmin}} \Big(a_{ij} \|z_{ij} - z_{ji}\|_2 + \frac{\rho}{2} (\|w_i^{t+1} - z_{ij} + u_{ij}^t\|_2^2 + \|w_i^{t+1} - z_{ji} + u_{ji}^t\|_2^2) \Big)$
  return z_{ij}^{t+1}, z_{ji}^{t+1}
procedure U-UPDATE(w_i^{t+1}, z_{ij}^{t+1})
  $u_{ij}^{t+1} \leftarrow u_{ij}^t + (w_i^{t+1} - z_{ij}^{t+1})$
  return u_{ij}^{t+1}
procedure RHO-UPDATE(r^t, s^t, ρ)
  if $r^t > \mu * s^t$ **then**
    $\rho \leftarrow \nu * \rho$
  else if $s^t > \mu * r^t$ **then**
    $\rho \leftarrow \rho / \nu$
  return ρ

[30] as the reference implementation of our Network-Lasso based machine-learning architecture for MEC-based IoT environments. SVM had already been widely used and shown to work well in prior trust research for developing classification- and regression-based trust prediction models [31], [32]. In fact, [33] proposes an SVM based classification model for predicting trustworthiness in IoT services as well, which aligns quite well with the primary scope of this study. This background provided us with a rational basis to adapt SVM as the local trust prediction problem to be solved as part of each sub-task running in the local MEC layers of the reference implementation. In that, each local MEC environment trains its own SVM-based binary classifier to predict untrustworthy IoT services. Each trained classifier classifies an input as either "benign" or "harmful" (denoted by "1" and "-1" respectively) indicating whether the IoT service in concern is trustworthy or not.

The task of obtaining the optimal separating hyperplane of the underlying soft-margin SVM, which separates the two classes that the classifier is trained for can be formally modelled as a minimization problem, as below.

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \epsilon_i. \\ &\text{s.t.} \;\; y_i(x_i^T.w + b) \geq 1 - \epsilon_i, \quad i = 1, ..., n \end{aligned} \tag{9}$$

where w denotes a weight vector corresponding to the model parameters to be learnt, x_i identifies the feature vector associated with i_{th} training sample fed into the learning model, y_i corresponds to the label associated with it, b is a bias while λ is a regularization parameter that determines the balance between widening the margin and ensuring x_i is classified accurately.

6 EVALUATION

This section is divided into four parts. Section 6.1 describes the experiments designed to evaluate the suitability of the proposed approach to address the challenges outlined in Section 1. Section 6.2 provides a technical summary of the state-of-the-art approaches that the proposed model was compared against while Section 6.3 briefly describes the datasets used. Finally, Section 6.4 shows and discusses the results of the experiments carried out. The source code of the proposed solution (depicted in Algorithm 2) and the simulations associated with the experiments is available in https://github.com/prabathabey/online-mec-trust.

6.1 Experiments

We conducted a series of experiments to comprehensively evaluate the effectiveness of the proposed approach to predict trustworthiness of real-time IoT services in MEC-based IoT systems. These experiments were aimed at justifying the ability of the proposed method to address the key challenges in predicting the trustworthiness of MEC-based IoT services outlined in Section 1 as well as comparing its performance against the state-of-the-art trust evaluation methods outlined in Section 6.2. The aforementioned experiments are organized into the following categories.

1) Ability of the proposed approach to derive the most relevant trust decisions under the presence of mobilizing service data providers and consumers (Challenge 1): This experiment focused on evaluating the adaptability of our approach to derive the most suitable trust decisions in the face of mobilizing IoT services and their consumers across neighboring MEC environments. The experimental results are presented and discussed in Section 6.4.1.

2) Effectiveness of the real-time and context-dependent trust prediction under heterogeneous environmental and operating conditions (Challenge 2): This experiment was designed to evaluate how the proposed approach predicts the trustworthiness of IoT services in a context-dependent manner. As part of it, we compared the results of our approach against both state-of-the-art trust prediction strategies that promote context-dependent trust prediction in a

distributed setting as well as centralized approaches that assume the entire MEC topology to be a single global context-environment for trust prediction. The evaluated centralized trust prediction approaches were used to train

- a single global model resembling a global trust prediction model trained in a typical centralized cloud environment with centrally accumulated trust information.
- a family of MEC-local trust prediction models atop locally accumulated trust information within each MEC environment resembling a set of non-communicative trust prediction models.

The results of these experiments are presented and discussed in Section 6.4.2.

3) Ability of the proposed approach to accurately derive trust decisions by sharing knowledge among neighboring MEC environments (Challenge 3): To assess this aspect, we compared the performance of the proposed model against that of a family of MEC-local trust prediction models trained atop locally accumulated trust information. These prediction models are non-communicative, and thus, by extension, not made to share knowledge amongst each other. The average accuracy returned by all MEC-local trust prediction models across the entire MEC topology in each approach was used as a key performance indicator to quantitatively assess and compare the performance returned by the evaluated approaches. The results of these experiments are presented and discussed in Section 6.4.3. To allow a fair analysis of the knowledge sharing ability of the proposed approach, only the baseline approaches that trained SVMs were evaluated and compared. To that end, the MEC-local non-communicative SVMs, global SVM, proposed approach with knowledge sharing enabled as well as disabled were compared.

Furthermore, each of the aforementioned experiments was also evaluated to assess the applicability into MEC-based IoT systems under the following key aspects.

4) Computational efficiency: We primarily considered *total running until the considered time horizon T elapsed* as the primary KPI of the computational efficiency. To that end, we have compared the total wall-clock time taken by the proposed approach as well as the other state-of-the-art trust prediction models until the time horizon T elapsed. The aforementioned metric is comprised of two key components in the form of 1) the total time taken to train the underlying trust evaluation model (in the context of predictive approaches) or evaluate a given trust decision, and 2) the time taken for the communication between the MEC layer and the centralized cloud layer (in the context of the distributed and collaborative approaches). The experimental results are presented and discussed in Section 6.4.4.

5) Communication efficiency: To evaluate this, the number of *rounds of communication* needed during the end-to-end process that includes trust information accumulation and prediction model training between the centralized cloud and distributed MEC layers was measured and analysed. Here, the aforementioned metric takes into account the number of times the data has been transmitted between the MEC environments and centralized cloud layer during the end-to-end process that spans across data accumulation as well as trust prediction model training. Therefore, it was assumed to be indicative of the network stress on core mobile networks of mobile network providers. The results of these experiments are presented and discussed in Section 6.4.5.

6) Scalability: This experiment was designed to evaluate the ability of our approach to realize the goals announced in 1 scalably as the size of the underlying MEC topology and trust information accumulated increases.

- To assess *the ability to scale well to growing topology sizes*, we monitored the average prediction accuracy across all the distributed trust prediction models in a given MEC topology and the average number of communication rounds required till convergence when the number of MEC environments in the underlying MEC topology is gradually increased. The other non-distributed state-of-the-art models were left out from this experiment as they used only a single model, which is either trained in a MEC-local or global manner. Consequently, these models do not scale across a given MEC topology.
- To assess *the ability to efficiently process large datasets accumulated across MEC environments*, we monitored and compared the total time taken to train the trust prediction models by each compared state-of-the-art trust prediction models.

The results of these experiments are presented and discussed in Section 6.4.6.

6.2 Compared Models

We compared our approach against a comprehensive set of state-of-the-art dynamic and/or online machine learning-based trust prediction models that honor the concept of trust evolution. A detailed summary of their implementations, simulations as well as the relevant assumptions and interpretations made while using them in our experiments is provided below.

CTRUST [14]: This approach shares the same notion of features contributing to trustworthiness of IoT services. Therefore, we seamlessly used the same trust features, upon which the proposed approach trains their corresponding MEC-based trust prediction models, to implement the trust evaluation model proposed in this particular work. As elaborated in Section 1, however, our solution has been derived on the assumption that IoT service consumers are less likely to communicate with each other. Therefore, the components in this particular approach that correspond to using reputation metrics to derive the indirect trust of an IoT service have been ignored. Two variants of this approach were implemented, one emulating a single global CTRUST model resembling one deployed in a centralized cloud environment, and a family of MEC-local CTRUST models resembling a non-communicative set of trust prediction models deployed within each MEC environment.

SC-TRUST [34]: This approach can be considered as an extended version of CTRUST described above, which focuses predominantly on service trust evaluation in the context service composition. Therefore, we cherry-picked the components related to the underlying trust evaluation model

used, and implemented in our experiments. This resulted in a trust evaluation model similar to that of CTRUST [14]. Similar to its predecessor, we implemented two variants of this trust evaluation model resembling its use within a centralized cloud environment and a non-communicative family of MEC environments in a given MEC topology.

Adaptive Trust [15]: We used this approach to establish a trust evaluation model for *a community of homogeneous services* within MEC environments. This approach uses a hybrid strategy to establish a trust evaluation model in which only the partial trust inference is done online, by default. However, to allow a fairer comparison, we attempted to do both trust evaluation model training and inference both in an *online* environment. In addition, since we focus only on homogeneous services, we also assumed that there is only one *usage* available, and therefore, omitted the usage-to-factor estimation. Two variants of this approach were implemented, one emulating a single global adaptive trust model resembling one deployed in a centralized cloud environment, and a family of MEC-local adaptive trust models resembling a non-communicative set of trust prediction models deployed within each MEC environment.

6.3 Datasets

For the experiments described above, we used multiple public IoT datasets in our simulations. A comprehensive overview of the structure of these datasets is given below.

N-BaIoT[4]: This dataset consists of network traffic data previously used to detect Mirai and BASHLITE attacks within an IoT setting [35]. Under each family of attacks, there were multiple individual attack types of which the records ($\in R^{115}$) were consolidated under the label *harmful*. In addition, the records related to legitimate network traffic ($\in R^{115}$) were classified under the label *benign*. The resulting dataset was normalized and split into 100 randomly-sized ($n \in [25000, 100000]$) splits simulating 100 MEC-local datasets.

WS-Dream[5]: This dataset consists of time-aware QoS data used for web service recommendation collected from 339 users against 5826 web services. This data has originally been collected at 15-minute interval over 64 timeslices. To generate a significantly sized dataset for our experiments, we transformed the original dataset where each tuple of <response time, throughput> as a single transaction between a user and service. This resulted in an aggregated dataset where each record in the dataset is of R^{109}) dimensionality, and altogether, there has been of 1974675 training examples available. The resulting dataset was normalized and split into 100 randomly-sized ($n \in [10000, 20000]$) splits simulating 100 MEC-local datasets.

6.3.1 Simulating continuous streams of trust information

To simulate a continuous stream of trust information arriving at each MEC environment, T randomly-sized mini-batches were drawn from each simulated MEC-local dataset sequentially and without replacement to simulate an incoming stream of arbitrarily-sized groups of trust information over a T number of timeslices. T here represents a finite time horizon, and was assigned to be 250 and 1000 for N-BaIoT and WS-Dream datasets, respectively. In addition, for simplicity, we assumed the quantity $h(t)$=1 indicating that each trust record used by the proposed approach for training the underlying trust prediction models carries similar weight.

All the simulations were developed in Python (v3.10/6), and carried out within an Amazon EC2 instance (c3.8xlarge) running Ubuntu 22.04 (64-bit) with 32vCPUs and 60GiB memory.

6.4 Results and Discussion

6.4.1 Effectiveness of real-time trust prediction under mobilizing IoT services and consumers

The results of our experiments confirmed that the proposed online and distributed trust prediction approach outperformed most state-of-the-art dynamic trust evaluation and the other baseline approaches evaluated, atop both N-BaIoT and WS-Dream datasets (see TABLE 1). However, the global SVM approach was observed to be 1.06% better in accuracy than the proposed approach atop N-BaIoT dataset, although the latter outperformed the former by 0.38% atop the WS-Dream dataset. These observations can be explained by pointing to the ability of the global SVM model to see a complete and an aggregated view of the concept of IoT service trust represented by the accumulated data, whereas MEC environments only see a split view of the world. Observed to be on part with the performance of the global SVM model in the aforementioned setting, we conclude that the performance of the proposed approach to be satisfactory. In addition, the superior communication efficiency of our proposed approach as discussed in Section 6.4.5 together with on-par performance with the global SVM approach can be deemed to make our approach more suitable to the problem setting announced in Section 1.

Furthermore, in comparison to the non-collaborative local SVMs, our approach was observed to be perform significantly better with a positive accuracy difference of 3% and 5.46% atop N-BaIoT and WS-Dream datasets. This could be attributed to the ability of the proposed approach to tackle the task of training a distributed family of MEC-local trust prediction models adhering to the context-specific trust characteristics also with the help of knowledge sharing. These aspects are analysed comprehensively in Section 6.4.2 and 6.4.3.

6.4.2 Effectiveness of real-time and context-dependent trust prediction under heterogeneous environmental conditions

The results of the experiments affirmed that the proposed approach is best suited to the problem at hand compared to most of the existing centralized as well as all the context-dependent trust evaluation strategies. In other words, the proposed approach outperformed most of the non-context-aware, centralized trust evaluation models by a considerable margin (see TABLE 1). In addition, it was observed that the proposed approach also outperformed the simulated

[4]https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[5]https://github.com/wsdream/wsdream-dataset

Model	N-BaIoT	WS-Dream
Local SVMs	83.9	50.26
Global SVM	87.96	55.34
CTRUST (Local)	82.28	50.19
CTRUST (Global)	82.28	59.19
SC-TRUST (Local)	82.28	50.19
SC-TRUST (Global)	82.28	50.19
Adaptive Trust (Local)	42.11	49.3
Adaptive Trust (Global)	54.12	52.37
Proposed approach	**86.9**	**55.72**

TABLE 1: Average prediction accuracy (%) of the evaluated models atop N-BaIoT and WS-Dream datasets on a simulated MEC topology containing 100 MEC environments.

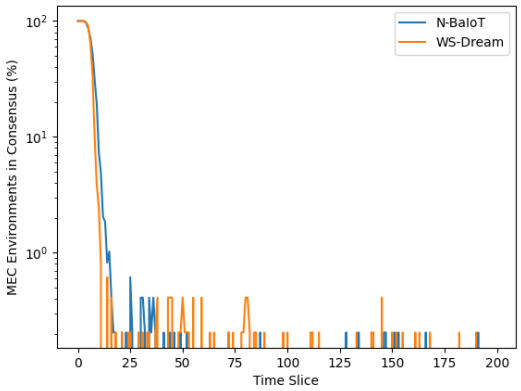

Fig. 3: Change in *percentage consensus* over progressive timeslices of the proposed approach atop N-BaIoT and WS-Dream datasets with a topology of 100 simulated MEC environments

context-aware trust evaluation strategies formed by running the existing centralized trust evaluation strategies within each MEC environment of a given MEC topology atop MEC-local datasets representing a family of contexts for trust prediction (see TABLE 1).

To further evaluate the finer characteristics of context-awareness supported by the proposed approach, we also observed the percentage of MEC environments *in consensus* in terms of their model parameters. In this setting, *MEC environments in consensus* reflect the similarity of their trained trust prediction models, and consequently, the similarity of their respective contexts for trust prediction. For this, we compared the MEC-local trust prediction approaches used in the evaluation against the proposed trust prediction strategy. These compared approaches included the local SVMs and the local variant of Adaptive Trust. The global variants of the aforementioned approaches were left out of the comparison as they implicitly consider a single global context-environment for trust prediction by design. Meanwhile, CTRUST and SC-Trust approaches too were omitted as they do not train any trust prediction models and only evaluate trustworthiness of services on-demand.

The results of this comparison revealed that the proposed approach was able to not only train a family of context-aware trust prediction models, but also identify similar context-environments for knowledge sharing. In Fig. 3, a *non-zero* value for the percentage of MEC environments in consensus indicates that *some* MEC environments produced similar trust prediction models whereas 100% indicates some similarity amongst *all* trained trust prediction models. In contrast, a *zero* percentage of MEC environments in consensus indicates that all 100 MEC-local trust prediction models trained by the 100 simulated MEC environments are different from each other. Such similarity corresponds to similar context-environments for trust prediction. As seen in Fig. 3, as the proposed approach progressed towards its time horizon T, the percentage consensus amongst MEC environments too indicated movement beginning with 100% down to smaller values, also reaching 0% at times. This corresponds to a higher degree of similarity at the beginning, and as individual MEC environments gradually learn the characteristics of their individual context environments, they also in turn, gradually establish their own context environments for trust prediction, towards the time horizon T. As we later discuss in Section 6.4.3, the ability of our approach to adaptively identify similar context environments also allows sharing knowledge with other similar neighboring MEC environments to train comparatively more generalizable trust prediction models.

6.4.3 *Effectiveness of knowledge sharing*

The average prediction accuracy recorded of the collaborative SVMs trained by the proposed approach and the non-collaborative local SVMs (i.e. trained by the proposed approach with knowledge sharing among MEC environments *disabled*) showed 2.36% and 0.72% higher accuracy against the N-BaIoT and WS-Dream datasets, respectively (see TABLE 2). In addition, in comparison the non-collaborative local SVMs trained independently from the proposed approach showed a difference in accuracy of 3% and 5.46% atop N-BaIoT and WS-Dream datasets, respectively. Given both the collaborative and non-collaborative SVMs were run under identical environmental settings and similar implementations, the above accuracy gain of the collaborative SVMs can be attributed to the effect of collaboration through knowledge sharing enforced by the proposed approach. Conversely, it could be deemed that MEC-local non-communicative trust prediction models trained by the state-of-the-art approaches could be overfitted on the MEC-local datasets, which the proposed approach was able to counter thereby allowing the collaborative SVMs to be more generalizable and achieve higher accuracies.

6.4.4 *Computational efficiency*

The results of our experiments to assess the computational efficiency of the proposed approach against the other baselines and state-of-the-art solutions are summarized in TABLE 3. As it is evident, *the total running time* taken for the proposed approach until the considered time horizon T elapsed was observed to be comparatively higher than most approaches evaluated. This could be attributed to the computational complexity of the underlying implementation arising from the solving of multiple sub-problems at a given

Model	N-BaIoT	WS-Dream
Local SVMs	83.9	50.26
Global SVM	87.96	55.34
Proposed model w/o knowledge sharing	**84.54**	**55.0**
Proposed model with knowledge sharing	**86.9**	**55.72**

TABLE 2: Average prediction accuracy (%) of the evaluated models atop N-BaIoT and WS-Dream datasets on a simulated MEC topology containing 100 MEC environments.

iteration of the algorithm as well as the overhead associated with knowledge sharing. In contrast, most other evaluated approaches employ a strategy where they solve a single optimization problem in an iterative manner within the course of their respective execution cycles. Notably, though, CTRUST and SC-TRUST approaches require significantly less time to run as, in our context, they only depend on default partial trust values that are data independent to compute trustworthiness. Consequently, although the computational time required is significantly less, the accuracy of the results produced is also significantly low (see TABLE 1), which makes them less suitable to our problem setting.

Model	N-BaIoT	WS-Dream
Global SVM	20.91	2.66
CTRUST (Global)	8.84	4.23
SC-TRUST (Global)	8.84	4.23
Adaptive Trust (Global)	25.2	8.54
Proposed approach	**15.29**	**3.31**

TABLE 3: The average running time observed (in seconds) per timeslice per trust prediction model with a MEC topology of 100 simulated MEC environments.

6.4.5 Communication-efficiency

The results of our experiments showed that the number of communication iterations across the core networks of mobile networking providers required by the proposed approach is 107.28 and 5.89 times less than that of the centralized and global state-of-the-art trust prediction approaches we used in our evaluation (see TABLE 4) atop N-BaIoT and WS-Dream datasets. This stems predominantly from the fact that all the other global and centralized trust evaluation approaches require trust information to be typically sent to cloud-based data centers across the core networks of mobile network providers. In contrast, the proposed approach processes trust information locally within each MEC environment and only send information (i.e. model parameters of each MEC-local trust prediction model) to the global knowledge aggregator running in a cloud-based data center once per timeslice per MEC environment. Consequently, amongst all the evaluated approaches, the proposed model could be deemed to cause the least amount of network stress on the core networks of mobile networking providers.

6.4.6 Scalability

Our experiments on the scalability yielded interesting results. From a prediction performance point of view, it was

Model	N-BaIoT	WS-Dream
Global SVM	2682033	588775
CTRUST (Global)	2682033	588775
SC-TRUST (Global)	2682033	588775
Adaptive Trust (Global)	2682033	588775
Proposed approach	**25000**	**100000**

TABLE 4: The number of *rounds of communication* required across the core networks until the the point of achieving the maximum accuracy evaluated in an MEC topology of 100 simulated MEC environments.

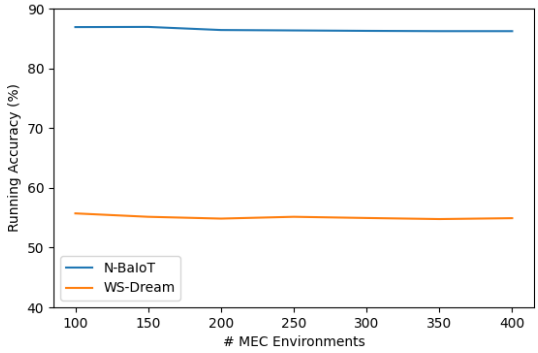

Fig. 4: Change in accuracy when the number of MEC environments in the underlying MEC topology is increased by 50 within the interval $[100, 400]$.

evident that the maximum accuracy produced by the proposed approach on top of both the datasets stayed relatively stable with minor movements when the number of nodes in the underlying MEC topology was increased (see Fig. 4). The aforementioned observation could be explained by referring to the fact that, at the point of reaching the considered time horizon T, the knowledge sharing among the underlying nodes would have transferred all shareable knowledge amongst the individual MEC environments to produce generalizable models at their respective maximum capacities. However, it was also seen a minor accuracy drop in all simulated topologies compared the one with 100 nodes in it. This can be attributed to the increasing tension that can at times be introduced when propagating knowledge over a given MEC topologies with the increasing number of MEC environments causing some MEC environments to adapt efficient yet slightly less optimal models at each timeslice of the algorithm. Choosing more optimal hyperparameter values for λ, ρ and γ_i based on the characteristics of the underlying MEC-local datasets and other relevant properties can potentially counter this effect.

Meanwhile, from a computational efficiency perspective, the computational time required until reaching the time horizon T was observed to be near-linearly increasing with the number of MEC environments (see TABLE 6). In other words, as the number of MEC environments in the underlying simulated MEC topology was gradually increased by 50 between the interval $[100, 400]$, the proposed approach required naturally more computational time till convergence

	No. of MEC environments						
Dataset	100	150	200	250	300	350	400
N-BaIoT	**86.9**	**86.93**	**86.41**	**86.34**	**86.27**	**86.22**	**86.22**
WS-Dream	**55.72**	**55.15**	**54.85**	**55.15**	**54.95**	**54.77**	**54.92**

TABLE 5: Average prediction accuracy (%) of the evaluated distributed trust prediction models atop N-BaIoT and WS-Dream datasets with the number of MEC environments in the MEC topology gradually increased.

	No. of MEC environments						
Dataset	100	150	200	250	300	350	400
N-BaIoT	**15.29**	**22.43**	**33.25**	**36.1**	**42.58**	**50.11**	**57.39**
WS-Dream	**3.31**	**5.53**	**7.03**	**9.46**	**10.44**	**16.24**	**14.57**

TABLE 6: Average computational time observed (in seconds) per timeslice N-BaIoT and WS-Dream datasets with the number of MEC environments in the MEC topology gradually increased.

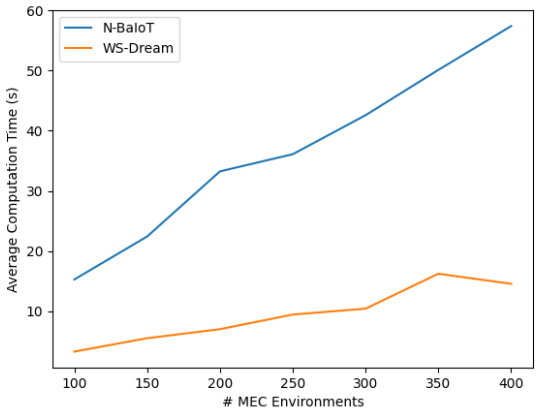

Fig. 5: Change in the computational time required until the considered time horizon T elapsed when the number of MEC environments in the underlying MEC topology is increased by 50 within the interval $[100, 400]$.

as evident in Fig. 5. This behaviour could be explained by referring to an interesting characteristic associated with the ability of the proposed approach to perform knowledge sharing. With an increasing number of MEC environments, the underlying knowledge sharing functionality could force the MEC environments to adapt sub-optimal models during different iterations of the underlying algorithm slowing down convergence.

7 Conclusions and Future Work

We proposed an edge intelligence-based strategy to predict the trust of the IoT service in real-time within the MEC-based IoT environment by using OADM. Specifically, we modelled the training of a set of distributed trust prediction models in an MEC-based IoT system as an online learning problem subjected to the dynamicity caused by the mobility of IoT services and their consumers as well as the heterogeneous operating conditions of different MEC environments that lead to multiple heterogeneous contexts for trust prediction. Further, we investigated making using of knowledge sharing across MEC environments to address the issues suffered by building a local trust prediction model by using merely the local data within an MEC environment. Our proposed method was evaluated in comparison to the state-of-the-art trust prediction approaches working in the centralized as well as distributed settings. The results verify the outstanding performance of the proposed method.

Our future work includes but is not limited to investigating the effective and efficient distributed hyperparameter tuning strategies for the proposed approach, incorporating lightweight dimension reduction techniques, e.g., [36], to further improve the computational efficiency of the proposed method, designing a method based on our previous work [37] for automatically finding the most suitable MEC topologies to pursue the best learning of the distributed prediction models, and conducting a theoretical analysis on the convergence of the proposed method.

Acknowledgments

This research was supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP220101823).

References

[1] P. Abeysekara, H. Dong, and A. Qin, "Machine learning-driven trust prediction for mec-based iot services," in *2019 IEEE ICWS*, 2019, Conference Proceedings, pp. 188–192.

[2] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[3] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*, Conference Proceedings, pp. 48–55.

[4] J. Daubert, A. Wiesmaier, and P. Kikiras, "A view on privacy and trust in iot," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, Conference Proceedings, pp. 2665–2670.

[5] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-based secure and trustworthy internet of things in sdn-enabled 5g-vanets," *IEEE Access*, vol. 7, pp. 56 656–56 666, 2019.

[6] P. Abeysekara, H. Dong, and A. Qin, "Data-driven trust prediction in mobile edge computing-based iot systems," *IEEE Transactions on Services Computing*, 2021.

[7] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1, p. 19, 2017.

[8] Y. Zhang, B. Di, P. Wang, J. Lin, and L. Song, "Hetmec: Heterogeneous multi-layer mobile edge computing in the 6 g era," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4388–4400, 2020.

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

[10] K. Zolfaghar and A. Aghaie, "Evolution of trust networks in social web applications using supervised learning," *Procedia Computer Science*, vol. 3, pp. 833–839, 2011.

[11] H. L. Nguyen, O.-J. Lee, J. E. Jung, J. Park, T.-W. Um, and H.-W. Lee, "Event-driven trust refreshment on ambient services," *IEEE Access*, vol. 5, pp. 4664–4670, 2017.

[12] C. Fernandez-Gago, F. Moyano, and J. Lopez, "Modelling trust dynamics in the internet of things," *Information Sciences*, vol. 396, pp. 72–82, 2017.

[13] Y. Alghofaili and M. A. Rassam, "A trust management model for iot devices and services based on the multi-criteria decision-making approach and deep long short-term memory technique," *Sensors*, vol. 22, no. 2, p. 634, 2022.

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

[15] M. Bahutair, A. Bougeuttaya, and A. G. Neiat, "Adaptive trust: Usage-based trust in crowdsourced iot services," in *2019 IEEE international conference on web services (ICWS)*. IEEE, 2019, pp. 172–179.

[16] B. Kantarci and H. T. Mouftah, "Mobility-aware trustworthy crowdsourcing in cloud-centric internet of things," in *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[17] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 international workshop on Self-aware internet of things*, Conference Proceedings, pp. 1–6.

[18] R. Neisse, G. Steri, G. Baldini, E. Tragos, I. N. Fovino, and M. Botterman, "Dynamic context-aware scalable and trust-based iot security, privacy framework," *Chapter in Internet of Things Applications-From Research and Innovation to Market Deployment, IERC Cluster Book*, 2014.

[19] S. Xu, Y. Qian, and R. Q. Hu, "Data-driven edge intelligence for robust network anomaly detection," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1481–1492, 2019.

[20] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, "Trust management and evaluation for edge intelligence in the internet of things," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103756, 2020.

[21] P. Zhou, S. Gong, Z. Xu, L. Chen, Y. Xie, C. Jiang, and X. Ding, "Trustworthy and context-aware distributed online learning with autoscaling for content caching in collaborative mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1032–1047, 2021.

[22] E. Gyamfi and A. Jurcut, "M-tads: A multi-trust dos attack detection system for mec-enabled industrial Iot," in *2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2022, pp. 166–172.

[23] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[24] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, and G. Verin, "Mec in 5g networks," *ETSI white paper*, vol. 28, pp. 1–28, 2018.

[25] H. Wang and A. Banerjee, "Online alternating direction method (longer version)," *arXiv preprint arXiv:1306.3721*, 2013.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[27] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, Conference Proceedings, pp. 387–396.

[28] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[29] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[31] J. Lopez and S. Maag, "Towards a generic trust management framework using a machine-learning-based trust model," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1, Conference Proceedings, pp. 1343–1348.

[32] R. Akbani, T. Korkmaz, and G. Raju, "Emltrust: an enhanced machine learning based reputation system for manets," *Ad Hoc Networks*, vol. 10, no. 3, pp. 435–457, 2012.

[33] U. Jayasinghe, G. M. Lee, T.-W. Um, and Q. Shi, "Machine learning based trust computational model for iot services," *IEEE Transactions on Sustainable Computing*, 2018.

[34] A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, X. Wang, and B. Zhou, "Sc-trust: A dynamic model for trustworthy service composition in the internet of things," *IEEE Internet of Things Journal*, 2021.

[35] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[36] A. K. Qin, P. N. Suganthan, and M. Loog, "Uncorrelated heteroscedastic lda based on the weighted pairwise chernoff criterion," *Pattern Recognition*, vol. 38, no. 4, pp. 613–616, 2005.

[37] A. K. Qin and P. N. Suganthan, "Initialization insensitive LVQ algorithm based on cost-function adaptation," *Pattern Recognition*, vol. 38, no. 5, pp. 773–776, 2005.

Prabath Abeysekara received his B.Sc. (Hons) of Engineering degree from University of Moratuwa, Sri Lanka in 2010. He is currently a PhD candidate at School of Computing Technologies in RMIT University, Melbourne, Australia. His primary research interests include Machine Learning, Cyber Security and Distributed Computing.

Hai Dong received a PhD degree from Curtin University, Australia in 2010. He is currently a senior lecturer at School of Computing Technologies in RMIT University, Australia. His primary research interests include: Service-Oriented Computing, Distributed Systems, Cyber Security, Machine Learning and Data Analytics. He is a senior member of the IEEE.

A. K. Qin received the BEng degree from Southeast University, China, in 2001, and the PhD degree from Nanyang Technological University, Singapore, in 2007. He is currently a full professor with Swinburne University of Technology, Australia. His major research interests include evolutionary computation, machine learning, image processing, GPU computing, and services computing. He is a senior member of the IEEE.