Expert Systems With Applications 238 (2024) 121676

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

*i*BUST: An intelligent behavioural trust model for securing industrial cyber-physical systems

Saiful Azad [a], Mufti Mahmud [b,c,d,*], Kamal Z. Zamli [e], M. Shamim Kaiser [f], Sobhana Jahan [a], Md. Abdur Razzaque [g]

[a] Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh
[b] Department of Computer Science, Nottingham Trent University, Clifton, NG11 8NS, Nottingham, UK
[c] Computing and Informatics Research Centre, Nottingham Trent University, Clifton, NG11 8NS Nottingham, UK
[d] Medical Technologies Innovation Facility, Nottingham Trent University, Clifton, NG11 8NS Nottingham, UK
[e] Faculty of Computing, University Malaysia Pahang, 26300 Gambang, Kuantan, Malaysia
[f] Institute of Information Technology, Jahangirnagar University, Savar, 1342 Dhaka, Bangladesh
[g] Department of Computer Science and Engineering, University of Dhaka, 1000 Dhaka, Bangladesh

ARTICLE INFO

Keywords:
Cyber security
Industry 4.0
Smart factory
FP-growth algorithm
Naïve Bayes

ABSTRACT

To meet the demand of the world's largest population, smart manufacturing has accelerated the adoption of smart factories—where autonomous and cooperative instruments across all levels of production and logistics networks are integrated through a Cyber-Physical Production System (CPPS). However, these networks are comprised of various heterogeneous devices with varying computational power and memory capabilities. As a result, many secure communication protocols – that demand considerably high computational power and memory – can not be verbatim employed on these networks, and thereby, leaving them more vulnerable to security threats and attacks over conventional networks. These threats can largely be tackled by employing a Trust Management Model (TMM) by exploiting the behavioural patterns of nodes to identify their trust class. In this context, ML-based models are best suited due to their ability to capture hidden patterns in data, learning and improving the pattern detection accuracy over time to counteract and tackle threats of a dynamic nature, which is absent in most of the conventional models. However, among the existing ML-based solutions in detecting attack patterns, many of them are computationally expensive, require a long training time, and a considerably large amount of training data—which are seldom available. An aid to this is the association rule learning (ARL) paradigm, whose models are computationally inexpensive and do not require a long training time. Therefore, this paper proposes an ARL-based intelligent Behavioural Trust Model (*i*BUST) for securing the CPPS. For this intelligent TMM, a variant of Frequency Pattern Growth (FP-Growth), called enhanced FP-Growth ($\mathcal{E}$FP-Growth) algorithm is developed by altering the internal data structures for faster execution and by developing a modified exponential decay function (MEDF) to automatically calculate minimum supports for adapting trust evolution characteristics. In addition, a new optimisation model for finding optimum parameter values in the MEDF and an algorithm for transmuting a 1D quantitative feature into a respective categorical feature are developed to facilitate the model. Afterwards, the trust class of an object is identified employing the Naïve Bayes classifier. This proposed model is evaluated on a trust evolution-supported experimental environment along with other compared models taking a benchmark dataset into consideration, where it outperforms its counterparts.

1. Introduction

In recent years, the industry-based manufacturing sector has faced several challenges, including short product lead times, diverse consumer needs, irregular demand fluctuations, and small-quantity batch production. To overcome many of these constraints, the Fourth Industrial Revolution or Industry 4.0 has emerged (Lee, Bagheri, & Kao, 2015). On the other hand, over the last decade, Artificial Intelligence (AI) and Machine Learning (ML) have been playing significant roles in the methodological developments in diverse problem domains,

* Correspondence to: Department of Computer Science, Nottingham Trent University, Clifton Campus, Clifton Lane, Nottingham NG11 8NS, UK.
E-mail addresses: saiful@cse.green.edu.bd (S. Azad), mufti.mahmud@ntu.ac.uk, mufti@ieee.org (M. Mahmud), kamalz@ump.edu.my (K.Z. Zamli), mskaiser@juniv.edu (M.S. Kaiser), sobhana@cse.green.edu.bd (S. Jahan), razzaque@du.ac.bd (Md.A. Razzaque).

https://doi.org/10.1016/j.eswa.2023.121676
Received 18 February 2023; Received in revised form 9 August 2023; Accepted 14 September 2023
Available online 25 September 2023

including computational biology (Rahman, 2018; Rakib, Rumky, Ashraf, Hillas, & Rahman, 2021), biomechanics (Zhang, Li, Xiao, & Zhang, 2023; Zhang et al., 2022), cyber security (Ahmed et al., 2021; Farhin, Kaiser, & Mahmud, 2021; Islam et al., 2021; Zaman et al., 2021), disease detection (Biswas, Kaiser, et al., 2021; Ghosh et al., 2021; Noor, Zenia, Kaiser, Mamun, & Mahmud, 2020; Wadhera & Mahmud, 2022a, 2022b, 2022c, 2023) and management (Ahmed, Hossain, et al., 2022; Akhund et al., 2018; Al Banna et al., 2020; Jesmin, Kaiser, & Mahmud, 2020; Mahmud et al., 2022; Sumi et al., 2018), elderly care (Biswas et al., 2021b; Nahiduzzaman et al., 2020), epidemiological study (Sadik et al., 2020), fighting pandemic (AlArjani et al., 2022; Bhapkar et al., 2021; Kumar et al., 2021; Mahmud & Kaiser, 2021; Paul et al., 2022; Prakash et al., 2021; Satu et al., 2021), healthcare (Mahmud, Kaiser, Hussain, & Vassanelli, 2018; Mahmud, Kaiser, McGinnity, & Hussain, 2021; Mahmud, Kaiser, et al., 2018; Nasrin, Ahmed, & Rahman, 2021; Rahman, Brown, Mahmud, et al., 2022), healthcare service delivery (Biswas et al., 2021a; Farhin, Kaiser, & Mahmud, 2020; Kaiser et al., 2021), natural language processing (Adiba, Islam, Kaiser, Mahmud, & Rahman, 2020; Das et al., 2021; Nawar et al., 2021; Rabby, Azad, Mahmud, Zamli, & Rahman, 2020; Rabby et al., 2018), social inclusion (Mahmud, Kaiser, & Rahman, 2022; Rahman, Brown, Shopland, Burton, & Mahmud, 2022; Rahman, Brown, Shopland, Harris, et al., 2022) and many more. To benefit from the current state-of-the-art AI-systems and towards creating Industry 4.0 compliant smart factories, the Cyber–Physical System (CPS) (Monostori et al., 2016; Nourian & Madnick, 2018) technology has appeared as the core infrastructural backbone on which a production pipeline is automated and leading to a Cyber–Physical Production System (CPPS) (Bicaku et al., 2017) that integrates autonomous and cooperative elements of the pipeline across all levels of production and logistics networks (Lee et al., 2015).

Since these networks connect the physical world or PW (i.e., communication-capable physical things) to the cyber world or CW (i.e., cloud computing, data analytics, communication, control platforms, etc.) and vice-versa, they experience notable threats and attacks as reported in a recent recommendation by the International Telecommunication Union (ITU-T Y.3052) (International Telecommunication Union, 2017). Of these threats and attacks, the PW may observe the installation and booting of fraudulent or modified software; the sensing systems may experience GPS spoofing, false signal injection, sensor device tempering, etc.; the CW may face eavesdropping, packet relaying, remote spying, and many more; finally, the core network may experience an impersonation of devices, traffic tunnelling between impersonating devices, intrusion, and many others.

Most of these threats and attacks can largely be tackled by employing a Trust Management Model (TMM), specifically by exploiting the behavioural or activity patterns of the nodes (Mahmud et al., 2019), and hence, the focus of this paper. It is noteworthy to mention that trust is a concept that can cover security and privacy aspects (International Telecommunication Union, 2017). Although several trust models are proposed for CPS and/or CPPS, but only a few of them are machine learning (ML)-based.

However, within this context, ML-based models are best suited due to their ability to capture hidden patterns in data, learning and improving the pattern detection accuracy over time to counteract and tackle threats of a dynamic nature, which is absent in most of the conventional models. Again, since trust evolves with time, ML-based models must adapt to this characteristic through a relearning process or any other appropriate method. On top of that, in a hostile environment where the likelihood of various threats and attacks are high with evolving facets, the adaptation capabilities of various ML-based models remain an important concern, which, to the best of our knowledge, has not been raised in the literature. To mitigate these issues, this paper aims to put forward an experimental procedure considering the learning, relearning and performance evaluation processes, which is designed and performed in a comprehensive manner incorporating the prominent trust attacks.

Also, among the existing ML-based solutions in detecting attack patterns, many of them are computationally expensive, require a long training time, and a considerably large amount of training data—which are seldom available (D'Angelo, Rampone, & Palmieri, 2017). An aid to this is the association rule learning (ARL) paradigm, whose models are computationally inexpensive and do not require a long training time (D'Angelo & Rampone, 2015; D'Angelo et al., 2017). Herein, Apriori (Wang & Zheng, 2020) and Frequent Pattern-Growth (FP-Growth) (Feng, Zhu, Zhuang, & Yu, 2018) algorithms are popular in mining rules (Borgelt, 2012) and are also suitable for this application. However, at times they fail to meet the required performance expectation due to the extraction of many irrelevant association rules (ARs) (Fournier-Viger et al., 2017). To overcome this issue, an updated version of the FP-Growth algorithm, named enhanced FP-Growth or $\mathcal{E}$FP-Growth, has been proposed in this paper with notably faster AR extraction time alongside making it suitable for the trust evaluation environment. Here, it is noteworthy to mention that in mining an appropriate amount of ARs relevant to a given context, minimum threshold support (minsupp) plays an important role. However, it requires searching an exponentially growing search space, which is a daunting task, especially considering the dynamic characteristic of the trust as well as attack evolution. To facilitate this process, a modified exponential decay function has also been developed to automatically calculate minimum support or minsupp for the $\mathcal{E}$FP-Growth algorithm.

Following the aforementioned discussions, this work has the following notable contributions:

i. A new and intelligent BTM is developed – named *i*BUST – using ML-based approaches with applicability to industry-based CPPS
ii. An existing algorithm is enhanced – called $\mathcal{E}$FP-Growth – by modifying the internal data structures of the FP-Growth algorithm for faster ARs' extraction
iii. A mathematical expression is modified – named Modified Exponential Decay Function (MEDF) – to calculate minsupp values automatically for aiding the adaptation of the $\mathcal{E}$FP-Growth algorithm to dynamically evolving trust characteristics
iv. An algorithm is designed for transmuting a 1D quantitative feature into a respective categorical feature employing the 1D K-means clustering and the Davis-Boulding (DB) indexing techniques, and
v. A trust evolution phenomena supported experimental setup is developed for incorporating prominent trust attacks during the learning and relearning processes and while evaluating the performance of the various models.

For the rest of the paper, the relevant trust models are detailed in Section 2, with Section 3 describing the network architecture. The proposed *i*BUST model is discussed in Section 4 with Section 5 contains the experimentation and the obtained results. A general discussion with the limitations of the proposed work and its future directions are mentioned in Section 6 and the work is concluded in Section 7.

2. Related work

Trust within a specified context refers to a firm belief in the ability of an entity to perform an action dependably, securely and reliably, and have been widely studied for a wide range of applications. Within the context of Information and Communication Technology (ICT) infrastructures and services, TMMs focus on solving security and privacy-related issues (Jayasinghe, Lee, Um, & Shi, 2019). Out of the available TMMs, only a subset of the existing security and privacy-related TMMs are applicable to CPPS as these models require careful design considerations related to both PW and CW. The development of models for trust management takes on many different attributes into consideration depending on the application paradigm (Yan, Zhang, & Vasilakos, 2014) and the popular models relevant to this context are discussed below.

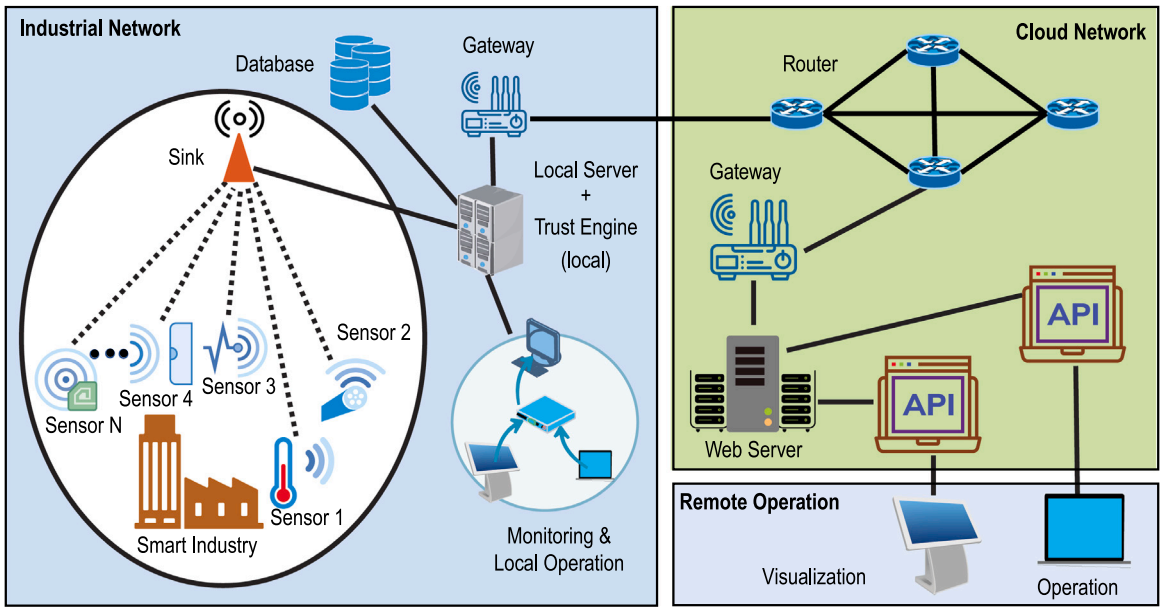

Fig. 1. The network architecture of a smart industry showing different parts of the CPPS.

Policy-based TMMs. The policy-based TMMs like (Gavriloaie, Nejdl, Olmedilla, Seamons, & Winslett, 2004; Nejdl, Olmedilla, & Winslett, 2004) estimate the trust class of an object depending on a set of predefined policies or rules. They are not suitable for dynamic and complex environments since they often rely on predefined perimeters.

Reputation-based TMMs. On the other hand, TMMs proposed in Blaze, Feigenbaum, Ioannidis, and Keromytis (1999), Xiong and Liu (2003) estimate trust classes based on reputation, i.e., by keeping track of the status of the interactions and behaviours. Alike Policy-based TMMs, they are also not suitable for dynamic and complex environments due to their inherent static nature.

Network-based TMMs. Furthermore, network-based models (Jayasinghe, Truong, Lee, & Um, 2016; Zhang, Chen, & Wu, 2006) employ a considerable amount of structural information like in-degree, out-degree and page rank concepts for extracting trust attributes or trust-relevant properties. Similar to previous two classes, they are not suitable for dynamic and complex environments due to lack of adaptation capabilities.

Game theory-based TMMs. In Pawlick and Zhu (2017), a game theory-based trust model is proposed for tackling Advanced Persistent Threats (APTs). However, the game theoretic approaches demand well-defined problems; whereas, in the real world, security-related problems are rarely well-defined and seldom static in nature. Similarly, several other conventional TMMs are proposed, including in Wang (2018), Zhao, Sun, Yue, Zhao, and Cheng (2018). However, an ML-based model is opted for in this paper for designing a new BTM for CPPS due to its capability of adapting dynamic and evolutionary characteristics of the trust as well as the attacks.

Fuzzy-based TMMs. Recently, a fuzzy-based brain-inspired model has been proposed in Mahmud et al. (2019) for securing data communication targeting Neuroscience applications. Since it has been developed taking a specific application into consideration – i.e., Neuroscience applications – it is not suitable for CPPS.

ML-based TMMs. Again, a support vector machine (SVM) based trust computational model is proposed in Jayasinghe et al. (2019) for IoT services along with the K-means algorithm and the Elbow method for clustering and labelling. Similarly, several other SVM-based models are proposed in Chen and Xu (2009), Han et al. (2019). However, none of them is tested on trust evolution-supported experimental environments incorporating prominent attacks. Therefore, in this paper, an SVM-based model is developed for investigating its performance against other models in hostile environments.

The work in D'Angelo and Rampone (2015), D'Angelo et al. (2017) is based on the Apriori ARL and Bayesian classification with an application to pervasive computing, which has several drawbacks compared to the FP-Growth algorithm (Al-Maolegi & Arkok, 2014). Particularly, the Apriori algorithm fails to load the whole database in memory when the database size is large and this requires an increased number of disk read operations as they scan the database in each iteration. Conversely, FP-Growth scans a database only twice irrespective of the number of itemsets in the database (Al-Maolegi & Arkok, 2014). Moreover, FP-Growth provides more relevant frequent itemsets than that of Apriori (Borgelt, 2012). Therefore, a variant of the FP-Growth algorithm is proposed in this paper for faster AR extraction and for adapting trust evolution.

3. Network architecture

A conceptual CPPS network architecture is shown in Fig. 1. It can be seen in the figure that diverse types of devices are installed in the network, including sensors, actuators, sinks, servers, gateways, routers, and operating and visualisation panels to facilitate automation. This network is comprised of two independent sub-networks, namely Industrial and Cloud networks (Asif-Ur-Rahman et al., 2019). Here, the industrial network supports local operations and the cloud network supports remote operations.

Most of the end devices in the industrial network transmit and receive data through the sinks and access points, which are subsequently connected to the server where the proposed trust estimating model or trust engine is installed. The key responsibilities of this trust engine are: to accept communication requests from the requester nodes, to identify the trust class of the requesting nodes, and to grant access to the ones found trustworthy. Here, the remote operations are performed through designated panels that communicate with the web server using Application Programming Interfaces (APIs). The web server also hosts data storage to store connection data and related activities, which are later shared using APIs. The communication between the gateways

and the end devices takes place over secure communication protocols, such as Hypertext Transfer Protocol Secure (HTTPS), Internet Protocol Security (IPSec), Secure Sockets Layer (SSL), and others making the cloud network inherently secure. This leaves the industrial network vulnerable and is the focus of this work. Again, it is a known fact that internal attacks on a network have more adverse effects on its performance than external attacks (Giandomenico & de Groot, 2018; Lord, 2017).

For further discussion, let us designate the trust engine as the Trustor, X, since it evaluates the trust of the other nodes in the network and the remaining nodes as Trustee, Y. The trust of X on Y can be built through an influence of a trusted third party, Z. These designations are in line with the definitions in the existing literature (Lin & Dong, 2017). In many real-world scenarios, such as in Banks, these three actors are often chosen to derive the trustworthiness of the Trustee, i.e., Y. For instance, for many banks, if an applicant wants to open a new account, it is necessary to submit a referral letter from the applicant's employer or a reference of any existing account holder. Here, the bank is X, the applicant is Y, and the organisation that is providing a referral letter or the existing account holder is Z. Considering these designations, the following assumptions are made in designing the proposed TMM:

- The evaluation of trustworthiness is a unilateral process, which is only performed by X (e.g., the bank).
- The evaluation depends on past interactions between X (e.g., the bank) and Y (e.g., the applicant) and a recommendation from a trusted third party, Z (e.g., the recommender).
- The trustworthiness of a new Y (e.g., a new applicant) is influenced by the recommendation or reputation of Z (i.e., the recommender).
- Decisions taken by X (e.g., the bank) in the relearning sessions are revised by experts until satisfactory performance is reached (e.g., re-evaluation of an account opening application of Y by X).
- Trust is asymmetric, i.e., even though Y trusts X, there is no obligation that X must trust Y (e.g., the trust relationship between bank, X, and applicant, Y).
- Trust may be transitive within a given context, but not in nature. For instance, when X trusts Z for a given context (i.e., new account opening) and Z trusts Y in the same context, then, X may trust Y for that context. However, it may not be true outside the context.
- Reputation and interaction data are given equal priority in deciding the proposed model's capability to handle reputation-based situations (e.g., an existing account holder's previous reputation and transaction history affect the present reputation).

It should be noted that, in modelling trust in artificial systems, the frameworks attempt to explain how trust is formed, maintained, and evaluated in various contexts. Admitting the fact that mimicking human trust 'as is' in an artificial system is an *NP-hard* problem. Therefore, for the sake of simplicity, each trust model makes some assumptions that are relevant to a specific scenario, which may vary from scenario to scenario with a certain degree of similarity to the human trust mechanism.

4. Proposed model: *i*BUST

Fig. 2 shows the conceptual design of the proposed TMM. It determines a node's trustworthiness through a number of steps that can be categorised in five stages: *i*. trust attribute extraction, *ii*. feature categorisation and attribute vector identification, *iii*. behavioural signature detection, *iv*. node signature generation, and *v*. trust classification. These stages are described in the following subsections.

4.1. Trust attribute extraction

Even though, a CPPS network generates a large amount of data; however, only a few of them can be directly utilised in the trust classification process. Again, the performance of a model is subject to the appropriate selection of trust attributes (TAs) and generally are accumulated by scanning various systems and custom-level log files along with other relevant data sources. In this paper, the following TAs are selected due to their relationship to the PW and CW and their influence on trust estimation:

4.1.1. Relative frequency of interaction (RFI)

In general, the interaction frequency refers to the number of interactions take place between the trustor and the trustee within a given unit of observation time (Zhang, 2001). In the context of the CPPS network architecture, only successful connection requests are counted in interactions, and thus, it favours trustworthy over untrustworthy nodes. Hence, the feature values of the RFI, $\mathcal{F}_{XY}^{RFI}$ can be calculated by Eq. (1) (Mahmud et al., 2019).

$$\mathcal{F}_{XY}^{RFI} = \frac{n_{XY}}{N} \tag{1}$$

where n_{XY} is the number of successful communication requests from the trustee Y to the trustor X over an epoch of t and N is the total number of successful requests received by the trustor X within that epoch.

4.1.2. Co-location relationship (CLR)

In a CPPS model, objects always remain in a relationship with their owner(s) (a.k.a., Owner Object Relationship or OOR) and therefore, the movement nature (static or dynamic) of OOR always influence the calculation of CLR (Jayasinghe et al., 2019). Again, since most of the IoT devices involve in the production system seldom shift their physical locations and hence, a decision boundary involving the distance from the trustor and the time spent within that vicinity must be taken into account in trust class estimation. If a node's continual spending time within the decision boundary surpasses the minimum time threshold before a connection request is performed, can be considered as a trustworthy node. Hence, CLR can be calculated as follows (Jayasinghe et al., 2019):

$$\mathcal{F}_{XY}^{CLR} = \frac{1}{dist(X,Y)} \frac{\mathcal{G}_X \mathcal{G}_Y}{\|\mathcal{G}_X\| \, \|\mathcal{G}_Y\|} \tag{2}$$

where, $\mathcal{G}_X$ and $\mathcal{G}_Y$ are the GPS coordinates of the trustor X and trustee Y, respectively and the symbol "$\|.\|$" is the norm. Here, cosine similarity between the two nodes is calculated using the second term, which is normalised by the first term, i.e., the geo distance factor, which can be calculated employing the algorithm discussed in Veness (2016). The geo distance factor is considered here since it calculates the distance with respect to actual earth surfaces in contrast to the naive euclidean distance calculation function.

4.1.3. Intimacy (I)

In the context of social relationships, the intimacy or relationship duration of interaction is an important feature in calculating the trust level. Following this, in the IoT ecosystem, the higher the duration of interaction between a trustee and a trustor, the higher the intimacy. Now, considering the total communication time between the trustee Y with the trustor X, τ_{XY} and the time difference between their initial encounter, $\tau_{XY}^{initial}$ and the final encounter, τ_{XY}^{final}, intimacy can be calculated as below:

$$\mathcal{F}_{XY}^{I} = \frac{\tau_{XY}}{\tau_{XY}^{final} - \tau_{XY}^{initial}} \tag{3}$$

This feature is important since it appreciates the long-term relationship between the trustee and the trustor.

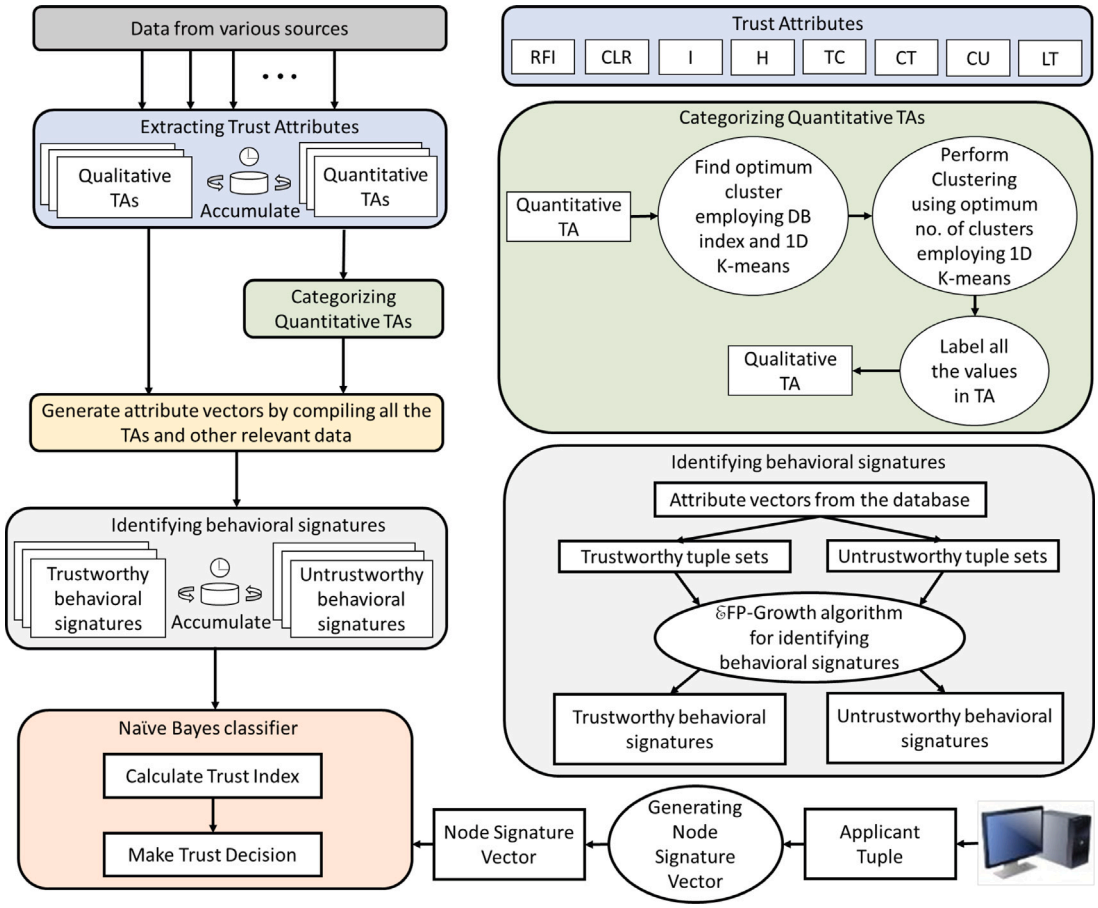

Fig. 2. The conceptual design of the *i*BUST TMM.

4.1.4. Honesty (H)

Analogous to social trust, honesty can be considered one of the main features for establishing trust between two IoT nodes in an IoT ecosystem. It can be determined using the successful and unsuccessful interactions between the involved nodes. While η^{s}_{XY} and η^{u}_{XY} denote successful and unsuccessful interactions between the trustor X and the trustee X, respectively. Hence, the feature values for honesty can be calculated as Mahmud et al. (2019):

$$\mathcal{F}^{H}_{XY} = \frac{\eta^{s}_{XY}}{\eta^{s}_{XY} + \eta^{u}_{XY}} \tag{4}$$

As can be observed from Eq. (4) is that the honesty value lies between $[0, 1]$, where 0 means no successful interaction occurs between the involved nodes, i.e., X and Y and 1 is the opposite.

4.1.5. Other features

Several other important features, which are reported in D'Angelo et al. (2017), are mentioned below:

Transactions Context (TC): This feature identifies the context of transactions, namely e-commerce, social networking, game, holiday, and others.

Counting Trust (CT): It is the count of trustworthy transactions belonging to a specific context that occur after encountering the last untrustworthy transaction.

Counting Untrust (CU): In contrast to CT, it is the count of untrustworthy transactions belonging to a specific context that occur after encountering the last trustworthy transaction.

Last Encounter (LT): It takes into account of the time when the identical context was last encountered.

4.2. Trust attribute categorisation and attribute vector identification

Among the aforementioned chosen TAs (see Section 4.1), except for transactions context, the rest of them are quantitative. However, ARL techniques are originally designed to work with discrete (categorical) TAs. Hence, in the subsequent sections, we discuss the relations between the quantitative and the categorical features followed by a technique of transmuting quantitative features to categorical features using our newly proposed algorithm.

4.2.1. Quantitative vs categorical features

As mentioned earlier, ARL techniques are originally designed to work with discrete (categorical) data or features. However, they can also support any continuous (quantitative) data or features by discretising them during the preprocessing stage. In detail, a quantitative feature can be partitioned into a number of consecutive intervals to generate new categorical values while preserving the initial order (D'Angelo & Rampone, 2015). Thereby, a quantitative rules problem is mapped into a Boolean rules problem, which is preferable for most of the ARL techniques (Srikant & Agarwal, 1996). However, this naive concept leads to two profound problems as reported in Srikant and Agarwal (1996): (*i*) a large number of partitions may not produce quality rules due to a small partition interval, called the "MinSup" problem and (*ii*) a smaller number of partitions experience possibility of losing information and confidence for a rule and hence, may not produce quality rules due to large partition interval, called the "MinConf" problem.

Algorithm 1: Clustering and labelling a feature

Input: $\mathcal{X} \leftarrow$ feature data, $\tau \leftarrow$ cluster threshold, $\delta \leftarrow$ DB_index, $\mathcal{F} \leftarrow$ feature name

Procedure *computeDBIndex* ($\mathbb{C}$, C)
- **for** β *in range* $|C|$ **do**
  - **for** i *in* $|C|$ **do**
    - $\psi_i \leftarrow$ calculate using Eq. (5) for cluster i
    - **for** j *in* $|C|$ **do**
      - $\psi_j \leftarrow$ calculate using Eq. (5) for cluster j
      - **if** $i \neq j$ **then**
        - $R_{ij} \leftarrow$ calculate using Eq. (6) for clusters i and j, respectively
        - R_{ij}_list.append(R_{ij})
  - $\sigma_R = \sigma_R + \max(R_{ij}\text{\_list})$
- $\delta = \sigma_R / |C|$
- return δ

Procedure *findOptimumNumberOfClusters* ($\mathcal{X}$, τ)
- **for** α *in range* (μ, τ) **do**
  - $\mathcal{L}$, $C \leftarrow$ kmeans1D ($\mathcal{X}, \alpha$)
  - $\mathbb{C} \leftarrow$ split into respective clusters
  - $\delta \leftarrow$ computeDBIndex($\mathbb{C}$, C)
  - δ_list.append(δ)
- $S \leftarrow$ index of min(δ_list) + μ
- return S

$S \leftarrow$ findOptimumNumberOfClusters ($\mathcal{X}$, τ)

$\mathcal{L}$, $C \leftarrow$ kmeans1D ($\mathcal{X}$, S) COMMENTS: Labelling feature values to respective clusters

for $\mathfrak{J}$ *in* $\mathcal{L}$ **do**
- file.write ($\mathcal{F}$ + "_range_" + $\mathfrak{J}$)

A solution for this conflicting set of problems is also proposed in Srikant and Agarwal (1996), where the authors combined base partitions with adjacent partitions while generating the rules. However, it leads to another problem, called the "ManyRules" problem. As the name suggests, the proposed solution generates many insignificant rules and thus, increases execution time. Again, to counter this ("ManyRules") problem, a solution using equal-depth partitioning is proposed in D'Angelo et al. (2017). However, this approach loses inter-cluster separability as well as intra-cluster homogeneity and compactness.

4.2.2. Clustering and labelling

For clustering and labelling quantitative features, an algorithm (Algo. 1) is developed in this paper for: (*i*) finding the optimum cluster size by integrating the 1D K-means clustering technique (Grønlund, Larsen, Mathiasen, & Nielsen, 2017) and the Davies–Bouldin (DB) indexing technique (Davies & Bouldin, 1979), (*ii*) identifying respective clusters of various feature values according to the optimum cluster size employing the 1D K-means clustering technique, and (*iii*) transmuting a quantitative feature into a respective categorical feature. In addition, the DB indexing technique is also amended for identifying the optimum number of clusters, S from a 1D data structure for a given feature.

First and foremost, the concept of DB_indexing is based on the intuition that a quality cluster requires a high inter-cluster separability as well as a high intra-cluster homogeneity and compactness. In this proposed model, every selected feature is partitioned into the optimum number of clusters since they facilitate in extracting the most relevant behavioural patterns. Here, a feature can be defined as below:

Definition 1. A feature, $\mathfrak{f} = \{\mathfrak{f}_1, \mathfrak{f}_2, \ldots, \mathfrak{f}_n\}$, where n is the number of members in $\mathfrak{f}$, and every, $\mathfrak{f}_i \in \mathfrak{f}$, where $1 \leq i \leq n$, is a feaure value.

Every $\mathfrak{f}$ can be divided into $k \geq 2$ clusters using 1D K-means clustering algorithm as demonstrated in Algo. 1, and denoted as $\mathbb{C}$.

Table 1
Snippet of a table that demonstrates an example of the RFI feature.

RFI Value	Cluster	Label
0.5895540503507924	2	RFI_range_2
0.1874549699881658	0	RFI_range_0
0.8281046274692764	3	RFI_range_3
0.10511064718072272	0	RFI_range_0
0.6540537102656461	2	RFI_range_2
0.3484650192862032	1	RFI_range_1
0.37024337793962736	1	RFI_range_1
0.665640115101046	2	RFI_range_2
0.8096313792134168	3	RFI_range_3
0.5565647719108703	2	RFI_range_2
...	...	...

Hence, it can have a set of k centroids, C in accordance to the number of clusters, i.e., $|C| = k$. Then, a cluster, $\mathfrak{C}_i$ can be defined as:

Definition 2. A cluster, $\mathfrak{C}_i = \{\mathfrak{f}_1^{\mathfrak{C}_i}, \mathfrak{f}_2^{\mathfrak{C}_i}, \ldots, \mathfrak{f}_m^{\mathfrak{C}_i}\}$, where m is the number of members in $\mathfrak{C}_i$, $|\mathfrak{C}_i| = m$, $m < n$, and every, $\mathfrak{f}_j^{\mathfrak{C}_i} \in \mathfrak{C}_i$, where $1 \leq j \leq m$, is a cluster member.

For calculating DB_index of $\mathbb{C}$, two measures are necessary to be calculated beforehand, namely ψ_i, which measures the dispersion of various points of any cluster, $\mathfrak{C}_i$ and R_{ij}, which measures the separation between two clusters, $\mathfrak{C}_i$ and $\mathfrak{C}_j$. Now, ψ_i can be found as follows:

$$\psi_i = \left(\frac{1}{|\mathfrak{C}_i|} \sum_{j=1}^{|\mathfrak{C}_i|} \left| \mathfrak{f}_j^{\mathfrak{C}_i} - C_i \right|^p \right)^{\frac{1}{p}} \tag{5}$$

where, C_i is the centroid of $\mathfrak{C}_i$ and since the values in the cluster are 1D; hence, the Euclidean distance function for 1D is employed to find the distance between two points, which is nothing but the absolute value of their difference. Again, ψ_i is the pth root of the pth moment of the points in $\mathfrak{C}_i$ about their mean. When $p = 1$, ψ_i becomes the average Euclidean distance of vectors in $\mathfrak{C}_i$ to the C_i; whereas, when $p = 2$, ψ_i is the standard deviation of the distance of samples in $\mathfrak{C}_i$ to the respective C_i.

Again, R_{ij} between the clusters $\mathfrak{C}_i$ and $\mathfrak{C}_j$ can be found as:

$$R_{ij} = \frac{\psi_i + \psi_j}{|C_i - C_j|} \tag{6}$$

where C_i and C_j are the centroids of $\mathfrak{C}_i$ and $\mathfrak{C}_j$, respectively. Again, since both the centroids are 1D, the Euclidean distance for the 1D function is employed here. If $R_i = max(R_{ij})$, where $i \neq j$, DB_index, δ can be calculated as:

$$\delta = \frac{1}{|C|} \sum_{i=1}^{|C|} R_i \tag{7}$$

This procedure of calculating δ for various k values iterates until $|C|$ as demonstrated in Algo. 1 and is also depicted in Fig. 3. Now, since only a limited number of clusters are usually generated, an exact method is preferable for identifying the optimum cluster size, S. Hence, a brute force method is employed to find S, which is the k that scores the lowest δ as could be seen in Fig. 3. As could be seen in the figure is that since cluster 4 scores the lowest DB_index, this is the optimum cluster size in this example. Afterwards, the final clusters are created employing S in 1D K-means clustering technique (see Algo. 1). Once clusters are identified for all the feature values, they are labelled concatenating the acronym of the feature, "_range_", and respective cluster id. A snippet of such a table for the RFI feature is presented in Table 1.

4.2.3. Attribute vectors identification

For modelling trust relationships between entities, both transmuted categorical features (see Section 4.2.2) and natural categorical features are combined together to form an attribute vector. In support of this

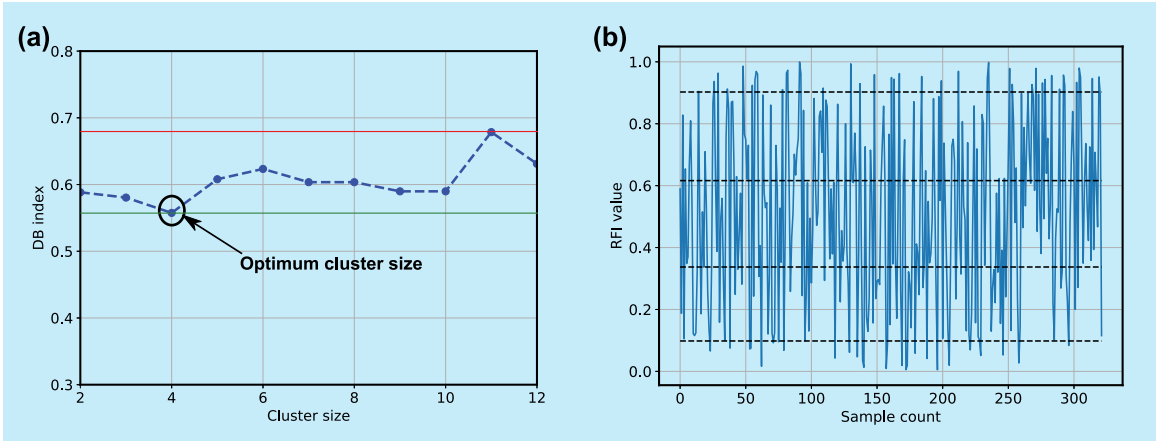

Fig. 3. Clustering with the acquired feature values using k-means clustering on 1D data and Davies–Bouldin indexing for finding optimum cluster size. (a) Finding optimum cluster size using the minimum Davies–Bouldin index. (b) Centroids distributions of the optimum cluster.

model, it can be argued that this model captures the context-based, identity-based, and recommendation-based trust relationships in an expressive and simple manner (D'Angelo et al., 2017). Later, these vectors are utilised by the Naïve Bayes classifier for identifying the trust class. An attribute vector (AV) can be defined as below:

Definition 3. $\mathfrak{E}_{ij}$ =<id$_j$, RFI, CLR, I, H, TC, CT, CU, LT>, where, $\mathfrak{E}_{ij}$ is an AV that records the experiences of a node i with another node j and id$_j$ is the unique identification number of node j, which is denoted by the network address.

These AVs are accumulated periodically considering new interaction data and stored in a database for all the interested nodes in the network for future use.

4.3. Detecting behavioural signatures using enhanced FP-growth

behavioural signatures are detected using an enhanced FP-Growth algorithm ($\mathcal{E}$FP-Growth) with the following enhancements as contributions of the current work.

4.3.1. Altering internal data structures

For extracting AR, the FP-Growth algorithm represents a database in the form of a tree data structure, called FP-tree, which excludes the necessity of candidate generation. Every node in this FP-tree represents an item from a record in the database. When a record, r is fetched from the database, the FP-Growth algorithm examines whether or not the prefix of r maps to a path in the tree. For the affirmative reply, the support counts of the corresponding nodes are increased. Conversely, new nodes are created and added to the tree with a support count of 1.

For accelerating the process of finding identical nodes, every node in the tree holds a reference to the next node. These connected nodes form a singly linked list for each item and the head of this list is stored in a table, called 'header table' as demonstrated in Fig. 4 (top). However, when a tree is considerably large with many subtrees due to the high variability of the items in the records of a database, it takes a long time in generating a suffix list using a singly linked list. The proposed enhancement replaces the header table and the singly linked list by offering the suffix list from a table, called 'link table' as demonstrated in Fig. 4 (bottom). Thereby, it reduces execution time for AR extraction and thus, respective association rule mining. Again, since the suffix list that is offered from the link table contains identical references like in the FP-Growth algorithm, the quality of the rules is not compromised for the proposed enhancement. In addition, the inclusion of a link table is not increasing space complexity with respect to its ancestor since it replaces the header table and the singly linked list which are of equal space.

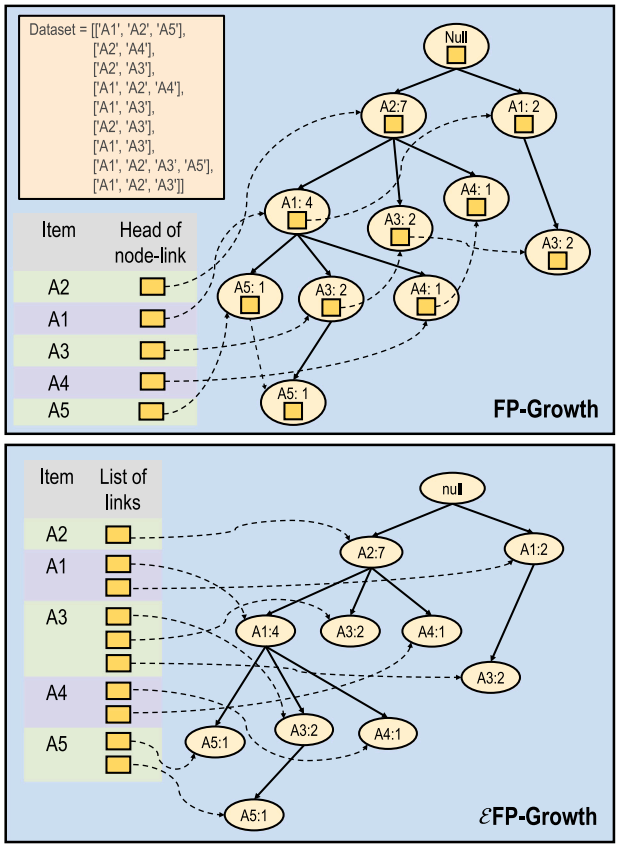

Fig. 4. Representation of the internal data structures of FP-Growth and $\mathcal{E}$FP-Growth algorithms.

4.3.2. Adapting trust evolution

The importance of selecting a suitable minsupp for extracting more relevant AR is reported in several literatures, including (Borgelt, 2012; Fournier-Viger, 2010). Generally, minsupp decreases with increasing dataset length (Fournier-Viger, 2010), and this threshold must be set with great caution. However, there is no really easy way to determine the best minsupp threshold. Again, in the case of a trust evolution environment where training data size increases over time, selecting a fixed minsupp threshold is impractical. Conversely, a formula for automatic minsupp calculation can be handy in this case. Therefore, a modified exponential decay function (MEDF) is developed from the

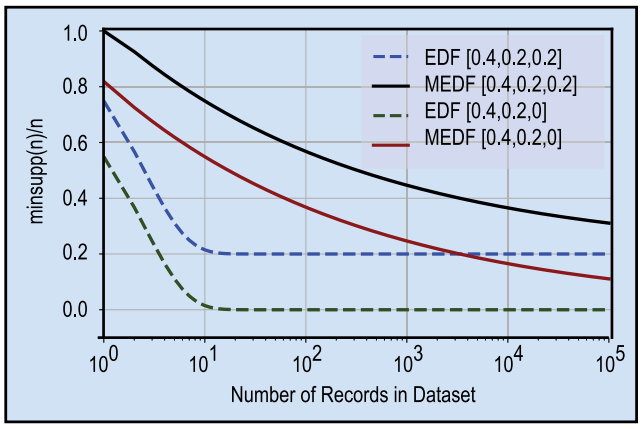

Fig. 5. Respective Y values of the exponential decay function and the modified exponential decay function for various c values with respect to various dataset lengths.

exponential decay function (EDF) proposed in Fournier-Viger (2010) as follows:

$$\text{minsupp}(n) = \left| e^{(-a\log_{10}(n)-b)} + c) \times n \right| \tag{8}$$

where, n is the number of records in the database and a, b, and c are positive constants, which contribute to the curve behaviour. The effect of these constants has been shown in Fig. 5, where it can be seen that the effect of constant c is not dominant in MEDF in contrast to EDF — even for a large value of n. Again, determining the values for constants, a, b, and c are tedious using a trial and error-based method, which is the most common method. Therefore, in this paper, an optimisation model is proposed for discovering optimum values for a, b, and c.

Algorithm 2: Pseudo Code for SCA Algorithm

Input: population $X = \{X_1, X_2, ..., X_n\}$, max_repetition (R), max_iteration (T), and constant magnitude (a)
Output: best outcome (X_{best}) and final population $X^f = \{X_1^f, X_2^f, ..., X_N^f\}$

```
for ρ = 1 till R do
    Initialisation;
    a ↩ 2;
    for t = 1 till T do
        Set initial r1 using Eq. (11);
        for all population in X^t do
            Evaluate each population, X_i^t by the objective function in Eq. (9);
            Update X_best based on the obtained result, if condition satisfied;
            Assign relevant random values to r2, r3, and r4 between [0, 1];
            Update X^t to X^{t+1} using Eq. (10);
return X_best and X^f
```

(a) An optimisation Model for Finding Optimum Parameter Values

Finding optimum values for the constants, a, b, and c in MEDF is an optimisation problem and demands an optimisation model. Moreover, since these constants can receive any value between the range $[0, 1]$, finding an optimum solution by any exact optimisation method within a linear time is infeasible (Zamli, Din, Nasser, & Alsewari, 2020). Conversely, meta-heuristic-based algorithms are preferable due to their adaptability to any optimisation problem and their ability to find a solution within a linear time. For this, the Sine-Cosine Algorithm (SCA) (Eyedali Mirjalili, 2016) has been chosen in this paper, which is a meta-heuristic algorithm with the added advantage of parameter independence. The SCA exploits the sine and cosine functions to perform both the local and global search by fluctuating outward or towards the global optimal solution, and hence, the name. It also introduces several random and adaptive parameters to facilitate the search process as presented in Algo. 2.

Since SCA is a population-based meta-heuristic algorithm, which generates population, X^t for iteration t where $t \in \mathbb{Z}_+$ encompassing N number of agents. Generally, agents are D-dimensional data structure; and hence, an agent (a.k.a candidate solution) can be described as $X_i^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\}$, where $1 \le i \le N$. In this paper, agents or candidate solutions are 3-dimensional that represent three parameters in Eq. (8), i.e., a, b, and c. From X^t, the best candidate solution, X_{best}^t is discovered using the following objective function:

$$x_{best}^t = \arg\min_{\ell^{X_i^t}} f(\ell^{X_i^t}) = \frac{\sum_{k=1}^{|\ell^{X_i^t}|}(h_k - \ell_k^{X_i^t})^2}{|\ell^{X_i^t}|} \tag{9}$$

where $\ell^{X_i^t}$ is a result ($\ell^{X_i^t} = 1$) or a list of results (when $\ell^{X_i^t} > 1$) that is acquired after experimenting with a certain metric using the candidate solution, X_i^t (or in other words, using the minimum support that is calculated using X_i^t); and h_k is the theoretically highest achievable result or target result for the instance or index k. The differences in the results are squared to support the minimum theoretical highest. The rationale of this equation is that it finds out how deviated the acquired result(s) is from the target result(s). Based on this equation, X_{best}^t offers the minimum deviation from the target. A practical implementation of this equation for a trust evolution phenomenon is explained in Section 5.

Afterwards, the best solution, X_{best} is updated if X_{best}^t satisfies the objective; and it is also utilised in updating the population for the next iteration, $X^{(t+1)}$, where the jth dimension of the ith candidate solution is found as follows:

$$X_{(i,j)}^{(t+1)} = \begin{cases} X_{(i,j)}^t + r_1 sin(r_2)|r_3 X_{(best,j)}^t - X_{(i,j)}^t|, & r_4 < 0.5 \\ X_{(i,j)}^t + r_1 cos(r_2)|r_3 X_{(best,j)}^t - X_{(i,j)}^t|, & otherwise \end{cases} \tag{10}$$

where r_2, r_3, and r_4 are random numbers ranges between $[0, 1]$. Again, since r_1 dictates the radius of the search circle, it can be adaptively and dynamically vary r_1 during the iteration process as follows:

$$r_1 = a\left(1 - \frac{t}{T}\right) \tag{11}$$

where T is the maximum iteration and a is the constant. After iteration t reaches T, the entire process is repeated with a different set of initialisation. This way, SCA keep repeating until the stopping condition is met or in other words, maximum repetition, R is encountered. The returned values, i.e., X_{best} and X^f, are utilised later in experiments; especially, X_{best} is utilised in calculating minimum support using Eq. (8).

4.3.3. Extracting behavioural signatures

To identify the trust (trustworthy (τ) and untrustworthy (υ)) behavioural signatures of a node, respective AVs are segregated into two respective sets. Each of these sets is passed to the $\mathcal{E}$FP-Growth algorithm to identify the associations among its items. The resulting associations constitute the behavioural signatures of that node for that particular TS class. Note that these signatures may change over time due to the dynamic nature of the trust; therefore, it needs to be revised after certain epochs.

4.4. Generating node signature vector

When the trust engine, X encounters any connection request from a node, Y, it updates $\mathfrak{E}_{XY}$. Afterwards, all possible frequent itemsets are extracted from $\mathfrak{E}_{XY}$. For example, assuming it is a three-dimensional vector containing $< \alpha_4, \beta_2, \gamma_1 >$, the possible itemsets that can be extracted from this vector are: $\{\alpha_4\}, \{\beta_2\}, \{\gamma_1\}, \{\alpha_4, \beta_2\}, \{\alpha_4, \gamma_1\}, \{\beta_2, \gamma_1\}$, and $\{\alpha_4, \beta_2, \gamma_1\}$. These itemsets are combined to build a node signature vector, ς_i, which is later passed to the Naïve Bayes classifier to determine its trust class.

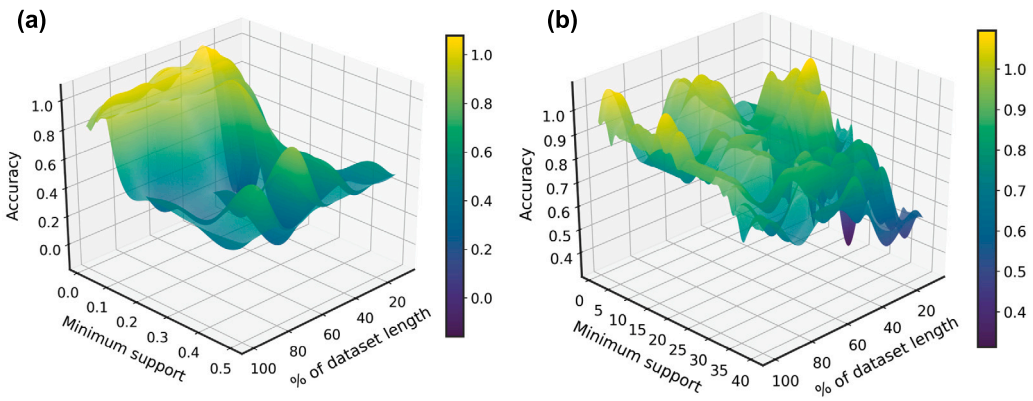

Fig. 6. Comparison between the surface graphs for accuracy for various lengths of dataset and minimum supports. (a) A surface graph that demonstrates the accuracy of the Apriori algorithm for DIUD with respect to various lengths of dataset and minimum supports. (b) A surface graph that demonstrates the accuracy of the FP Growth algorithm for DIUD with respect to various lengths of dataset and minimum supports.

4.5. Making decision using Naïve Bayes classifier

The Naïve Bayes classifier is utilised in this work for decision-making on a requested connection from a node. It is a Bayes theorem based on the probabilistic classifier with strong independence assumptions (Wang & Lin, 2019) that classifies the extracted ς_i (see Section 4.4) to the suitable TS class.

Assuming, h be a hypothesis, which says that ς_i belongs to a specific TS class, $P(h)$ is the prior probability of h, and $P(\varsigma_i)$ is the prior probability of ς_i. Hence, the probability that h holds for the observation ς_i, i.e., $P(h|\varsigma_i)$ can be found as:

$$P(h|\varsigma_i) = \frac{P(h)P(\varsigma_i|h)}{P(\varsigma_i)} \tag{12}$$

where, $P(\varsigma_i|h)$ is the posterior probability of ς_i given h, which can be calculated as:

$$P(\varsigma_i|h) = \prod_n P(\varsigma_{i_n}|h) \tag{13}$$

Now, let h be the TS class (trustworthy, untrustworthy), i.e., $h = \tau or \upsilon$ and $t \in h$; and $\varsigma_{i_n} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}$ be the node entity vector with n number of itemsets; then, Eq. (12) can be written as for a specific t:

$$P(t_{t\in h}|\varsigma_i) = \frac{P(t)\prod_n P(\varsigma_{i_n}|t)}{P(\varsigma_i)} \tag{14}$$

The obtained probability values for various t can be utilised to take the final decision, δ, which mentions the class where ς_i belongs. Again, since the denominator is identical for any t in Eq. (14), δ can be found as follows:

$$\delta(\varsigma_i) = \arg\max_{t\in h} P(t)\prod_n P(\varsigma_{i_n}|t) \tag{15}$$

Here, $P(\varsigma_{i_n}|t)$ is calculated using the Laplace estimator (Sarkar, 2019) to avoid the zero probability condition, which is probable in our scenario, as below:

$$P(\varsigma_{i_n}|h) = \frac{\sigma_{\epsilon_k} + 1}{\sigma + |d|} \tag{16}$$

where, σ is the sum of occurrence of the itemsets in the set of tuples that are associated with class t, σ_{ϵ_k} is the frequency of ϵ_k (where $k = 1, 2, \ldots, n$) and $\epsilon_k \in \varsigma_i$ in the set of tuples that are associated with t, and $|d|$ is the dimension of the tuples including all the categories.

As could be observed from Eq. (15) is that the class of ς_i is decided based on the highest probability value between $P(\tau|\varsigma_i)$ (trustworthy) and $P(\upsilon|\varsigma_i)$ (untrustworthy). For instance, if $P(\tau|\varsigma_i) = 4.86^{e-20}$ and $P(\upsilon|\varsigma_i) = 7.41e^{-20}$; since the latter scores the highest probability value, hence, the final decision is $\delta = \upsilon$ or untrustworthy.

5. Experimental evaluation

This section is divided into three subsections. Among them, the first subsection describes the experimental setup that has been taken into account during conducting the experiments, and the other two subsections compare the performance of the $\mathcal{E}$FP-Growth and the proposed model.

5.1. Experimental setup

In the following sections, the experimental setup is detailed.

5.1.1. Implementation details

For comparing the performance of the proposed BTM model (*i*BUST), two other models are considered, namely BTM with Apriori (BTM_Apriori) as in D'Angelo et al. (2017) and BTM with SVM (BTM_SVM) as in Jayasinghe et al. (2019). Since *i*BUST and BTM_Apriori are the ARL-based algorithms, both extract relevant behavioural signatures before classifying by the Naïve Bayes. On the contrary, since SVM is an association learning-based model, it is trained before classification.

All the compared BTMs are implemented using python and its relevant packages, including pyfpgrowth, random, Scikit-learn, apyori, matplotlib, and csv (Van Rossum & Drake, 1995). All implementation codes along with their respective datasets are currently available in Bllagdham and Azad (2020) to access upon request.

5.1.2. Datasets

The key dataset of this experimental evaluation is the Dishonest Internet User Dataset (DIUD) as in D'Angelo et al. (2017), which is a benchmark dataset for evaluating trust models. In addition, to compare $\mathcal{E}$FP-Growth with its ancestor, FP-Growth, two benchmark datasets are chosen, namely Kosarak and Accident (Goethals, 2004).

5.1.3. Performance metrics

The performances of the compared techniques are evaluated employing the following four performance metrics, which are calculated in accordance with the standard definition, also available in D'Angelo et al. (2017), Jayasinghe et al. (2019), namely Execution Time (ET), Accuracy (Ac), Sensitivity (Se), and Area Under ROC curve (AUC).

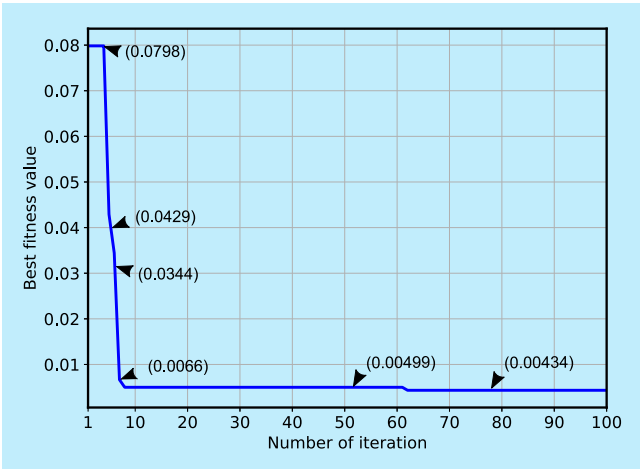

Fig. 7. Convergence curve for finding the best parameter values for various iterations.

Table 2
Summary of experimental setup.

Parameter	Options
Compared Models	*i*BUST, BTM_Apriori, BTM_SVM
Datasets	DIUD, Kosarak, Accident
Metrics	Execution Time, Accuracy, Sensitivity, Area Under ROC curve (AUC)
Indirect Attacks	Ballot Stuffing (BS), Bad Mouthing (BM), Random Opinion (RO)
Direct Attacks	Counting (CN), Context (CX), Timing (TI)
Relearning Intervals, M	10
Optimum values for Eq. (8)	$a = 7.91315688e-01$, $b = 8.45780180e-01$, $c = 4.87325936e-04$

5.1.4. Attacks

For the experiments, three nodes are selected, namely a trustor (X), a trustee (Y), and a trusted third party (Z). Note that since the process is similar for every node in the network, the inclusion of more nodes in the experiments will not influence the performance. Again, to make the learning (or training) and relearning (or retraining) process hostile for X, 3 indirect attacks are considered (D'Angelo, 2019), namely Ballot Stuffing (BS), Bad Mouthing (BM) and Random Opinion (RO). On top of that, to stress the effect, both attacks tamper up to 50% of the reputation data. Furthermore, 3 direct attacks, namely Counting (CN), Context (CX), and Timing (TI), which Y directly initiates while maintaining a good reputation with X, are considered (D'Angelo, 2019).

5.1.5. Supporting trust evolution

As it is a well-known fact that trust evolves with time, and hence, an efficient trust model should take into consideration during the design. In our case, it is performed by automatically calculating minsupps using MEDF. As mentioned earlier in Section 1 is that even though several ML-based trust models are proposed in the literature, their performances are hardly evaluated in a trust evolution scenario. Therefore, in this paper, a such scenario has been designed for evaluating the performance of all the compared models. For that, initially, the chosen dataset is divided into M segments for M relearning intervals in chronological order. Afterwards, during the relearning process, the current segment is merged with all the previously accounted segments imitating trust evolution.

5.1.6. Optimum parameter values selection

For finding optimum parameter values for all the compared models, $M = 10$ relearning intervals are considered. In other words, during the experiments, $f(x) = \{10x \mid 1 \leq x \leq 10,\ x = x + 1\}\%$ length of dataset is employed here. Again, 70% of the records are utilised for training and the rest (30%) of them are utilised for testing.

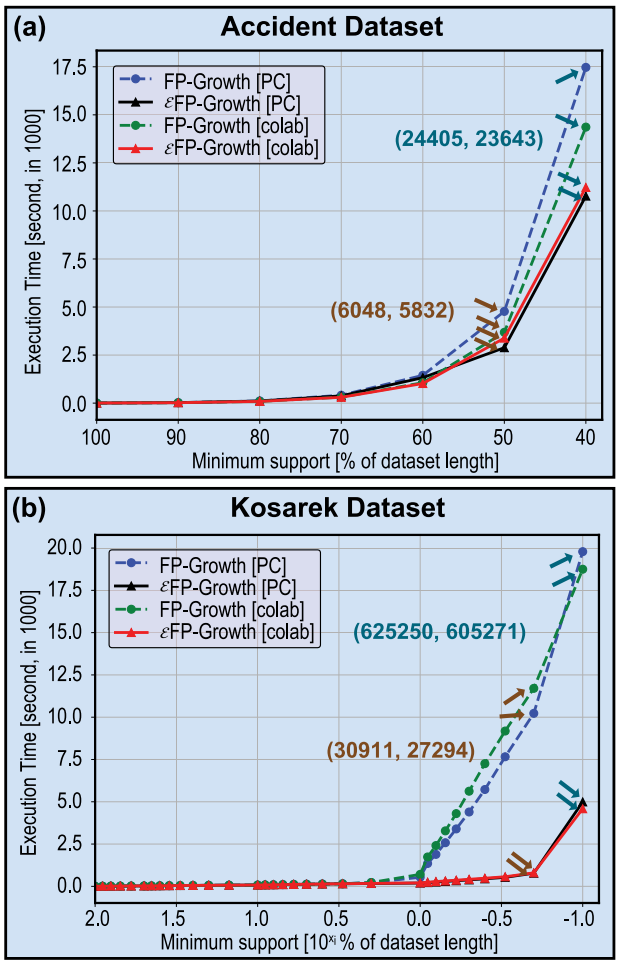

Fig. 8. Comparison between FP-Growth and $\mathcal{E}$FP-Growth. Here, the tuples present extracted frequent pattern count and association rule count, respectively.

Apriori. The accuracy results acquired for various minsupp with respect to various dataset lengths are plotted in Fig. 6 (a). As could be observed from the figure, although for some higher minimum support and for certain dataset lengths, some improvement in accuracy is observed; however, for any dataset length, the highest accuracy is achieved for the minsupp lower than 0.1. More specifically, it is received at 0.08 and hence, in BTM_Apriori, it is selected as the minsupp. Again, this optimum value is also in line with trust evolution since it is fixed for sequentially increasing dataset length. Note that since analogous observations are recorded for other metrics, they are not reported in this paper.

Enhanced FP-growth. Alike Apriori, the accuracy results of various minsupp with respect to various dataset lengths are plotted in Fig. 6 (b). However, unlike Apriori, the highest accuracy values are scattered throughout the entire experimental space. As could be observed from the figure is that for 100% length of dataset, minsupp 16 is providing the highest accuracy; whereas for 50% length of dataset, it is between [6, 10], and so on. Hence, we cannot fixed any minsupp like Apriori for $\mathcal{E}$FP-Growth, which would perform better for any length of dataset.

This observation also advocates the necessity of calculating minsupp automatically, which is performed in this paper using Eq. (8). Considering the identical scenario and metric, and $h_j = 1, \forall j$ in Eq. (9), the SCA return following optimum values: $a = 7.91315688e-01$, $b = 8.45780180e-01$, and $c = 4.87325936e-04$, which is also mentioned in Table 2.

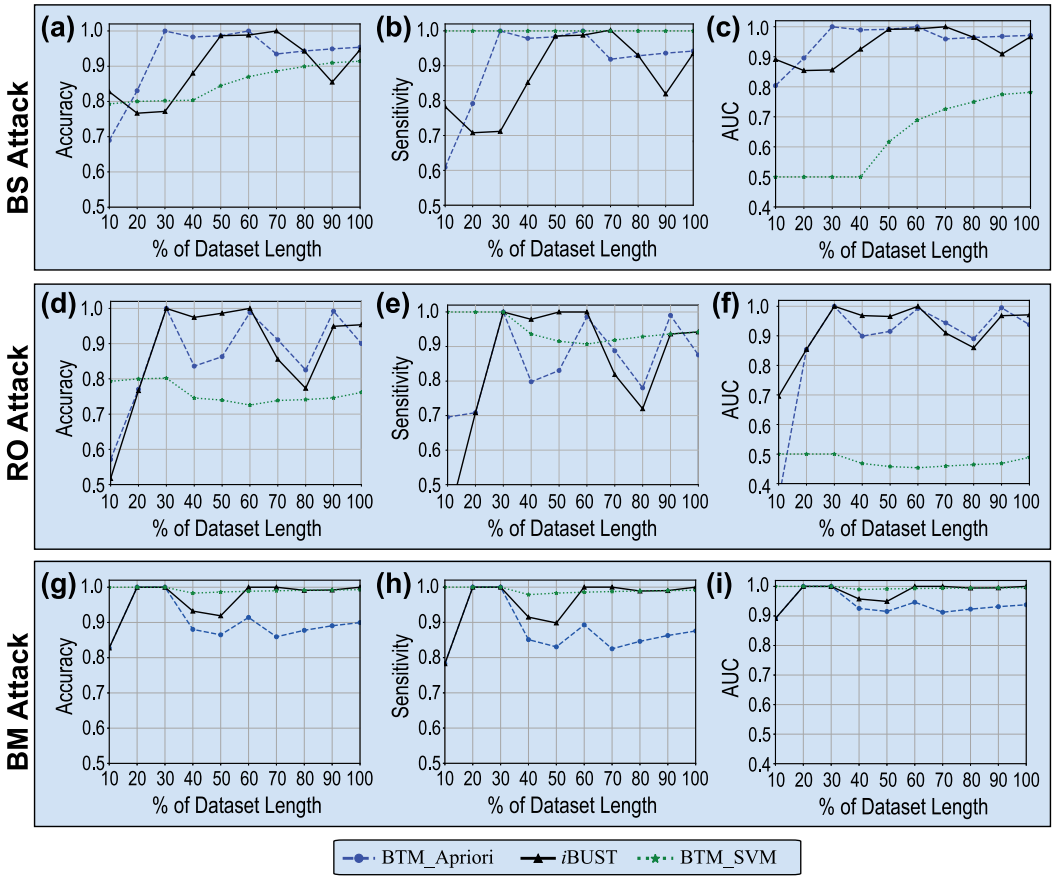

Fig. 9. Comparison of *i*BUST with other models for the CN attack against various indirect attacks.

Table 3
Parameters provided in GridSearchCV.

Parameter	Options
kernel	'linear', 'poly', 'rbf', 'sigmoid'
C	$\{c \mid 1 \leq x \leq 52, x = x + 1\}$
degree	$\{d \mid 3 \leq d \leq 8, d = d + 1\}$
coef0	$\{cof \mid 0.001 \leq cof \leq 10, cof = cof + 0.5\}$
gamma	'auto', 'scale'

The convergence curve for various numbers of iterations is depicted in Fig. 7 where returned values by the SCA are shown explicitly with arrows. As could be observed from the figure is that X_{best} is found after 62 iterations.

Support vector machine (SVM). It is always a difficult task in SVM to stipulate values for hyperparameters since the performance of SVM is outrightly influenced by the values specified for these parameters. Thanks to the GridSearchCV of the sklearn library (Pedregosa et al., 2011), which automates this process. It employs the grid search technique, and hence, the name. It requires a list of parameters and the range of values for each parameter. The range of values that are taken into account for various parameters in this paper are mentioned in Table 3.

In the case of GridSearchCV, a cross-validation process is performed in order to determine the hyperparameter value set which provides the best accuracy levels. The optimum parameters that are returned from it are mentioned below: {C=1, coef0=10, degree=3, gamma='scale', kernel='poly'}.

5.2. Performance comparison of enhanced FP-growth vs FP-growth

To compare $\mathcal{E}$FP-Growth with its ancestor, FP-Growth, two benchmark datasets are chosen, namely Kosarak and Accident (Goethals, 2004). In favour of our selection, we would like to argue that the AV list may keep increasing in a trust evolution environment over time; and a preferred algorithm must extract ARs within a short period of time irrespective of the length. In addition, two different environments — (i) Google colab pro or colaboratory pro and (ii) a PC with the specification – Intel(R) Core(TM) i7-4600U CPU @ 2.10 GHz 2.69 GHz, 8 GB RAM, 64-bit Windows 10 Pro operating system – are selected for performing the experiments. The justification for choosing two environments is that even though the execution times on a PC may vary due to running other background programmes and/or parallel programmes at that time; however, in a cloud-based environment like the colab pro, these constraints are generally absent.

The execution times for both the databases against different minimum supports are plotted in Fig. 8a and 8b. It can be observed from the figures is that, execution times of both the algorithms for both the datasets increase with the decreasing number of minsupp. It happens due to adding more number of nodes in the tree. As a result, a greater number of frequent patterns are extracted, and hence, a greater number of ARs are generated. Among the compared algorithms, $\mathcal{E}$FP-Growth elapses several magnitude lower execution time than its ancestor for lower minsupp for both the datasets as reported in Table 4. Thanks to the link table that provides suffix lists instantly from the table unlike FP-Growth, which has to generate these suffix lists employing the header table and the respective singly linked lists. Since a significant performance improvement is observed for the proposed $\mathcal{E}$FP-Growth algorithm, it is selected to conduct the rest of the experiments.

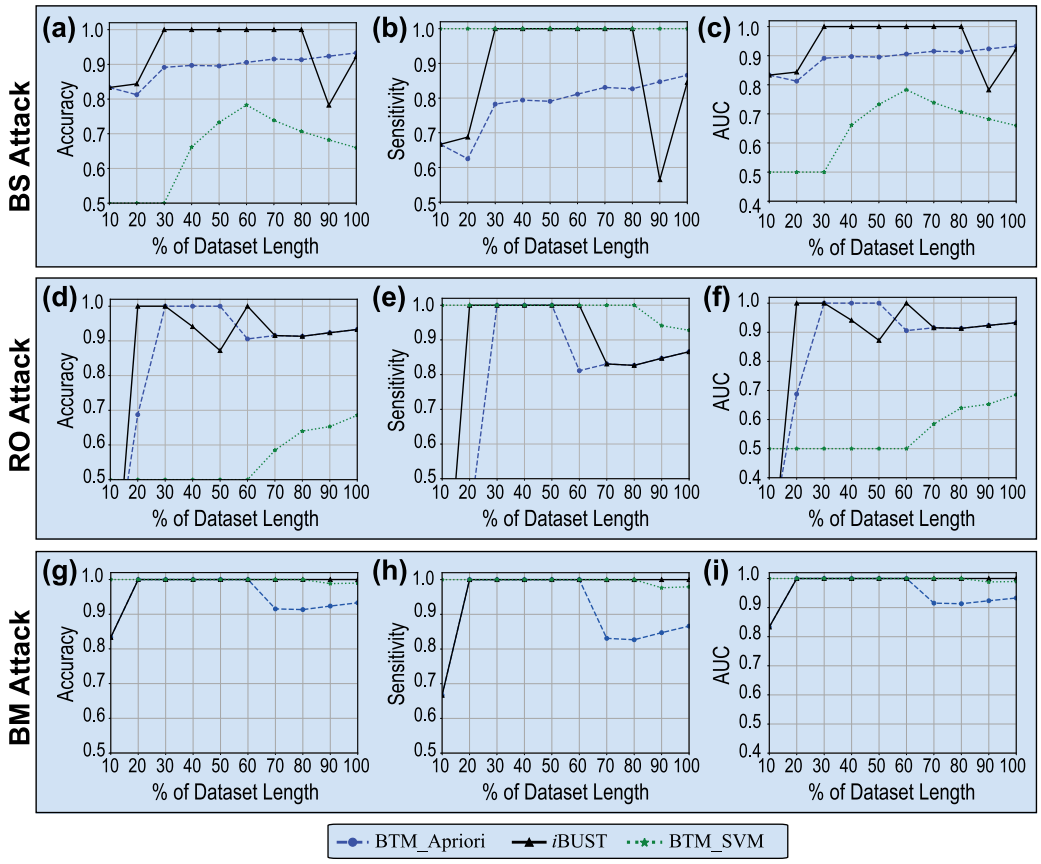

Fig. 10. Comparison of *i*BUST with other models for the CX attack against various indirect attacks.

Table 4
Execution times for the lowest minsupp for two datasets, two computing environments, and two algorithms as reported in Fig. 8.

Dataset	Minimum support	Algorithm	Computing environments	Execution Time (sec)
Kosarak	990	FP	PC	19 788.26
		FP	colab pro	18 749.25
		$\mathcal{E}$FP	PC	4992.17
		$\mathcal{E}$FP	colab pro	4587.88
Accidents	136 073.2	FP	PC	17 470.51
		FP	colab pro	14 361.11
		$\mathcal{E}$FP	PC	10 772.29
		$\mathcal{E}$FP	colab pro	11 222.82

5.3. Performance of BTM

Since trust evolves with time; it must be reflected in an experimental design when evaluating the performance of a model. Consequently, for capturing this phenomenon in our experiments, a cumulative approach is utilised where the selected dataset is divided into M segments representing M relearning (for $\mathcal{E}$FP-Growth and Apriori) or retraining (for SVM) intervals, which is set to 10 in this paper. Here, each chronological segment is merged with other explored segments before initiating a new round of relearning or retraining process, and thus, the length of the dataset increased and the models evolve with time. For the other experimental design issues, it follows the assumptions that are mentioned in Section 3.

5.3.1. Performance under CN attack

In this attack, for maintaining a good reputation, Y transmits 3 to 5 trustworthy requests consecutively followed by one untrustworthy request, and this cycle continues. The performance of the compared models is evaluated under 3 indirect attacks, namely BS, RO, and BM for 3 performance metrics: Ac, Se, and AUC; and the results are plotted in Fig. 9a to 9i. From the figures, it can be observed that the BTM_SVM model is severely affected by both the indirect attacks for Ac and AUC metrics. For BS, it suffers due to multiplicate entries in the dataset, which reduce the number of unique records; therefore, SVM is unable to discover the most representative hyperplane for separating the trust classes. On the other hand, for RO attack, it suffers since random opinion sometimes assigns trustworthy entries to untrustworthy entries, and vice versa; thereby, making it difficult for the BTM_SVM to discover a hyperplane that can precisely differentiate the classes. Same observation is true in case of BM attack where trustworthy entities are selected as untrustworthy entities due to negative or false information disseminated by the other colleagues.

On the other hand, for BS, the lowest performances for Ac and Se are received by the BTM_Apriori when the dataset length is 10%, 0.690 and 0.609, respectively; whereas, for AUC, BTM_SVM receives the lowest performance of 0.5 for the lower volumes of dataset, i.e., 10% to 40%. On the other hand, for RO, $\mathcal{E}$FP-Growth attains the lowest performances for the lowest volume of dataset, which are 0.517, 0.391, and 0.696 for Ac, Se, and AUC, respectively. However, with increasing direct interactions, the performance of $\mathcal{E}$FP-Growth recovers and in several instances overpowers other compared models, which shows its capability of trust evolution even in hostile environments.

On the other hand, both *i*BUST and BTM_Apriori exhibit a comparative performance in case of RO. Although, they struggle a bit at the beginning, but they recover later with more direct interaction data between X (trustor) and Y (trustee). It is because, with more direct interaction data, they were able to find out more accurate behavioural

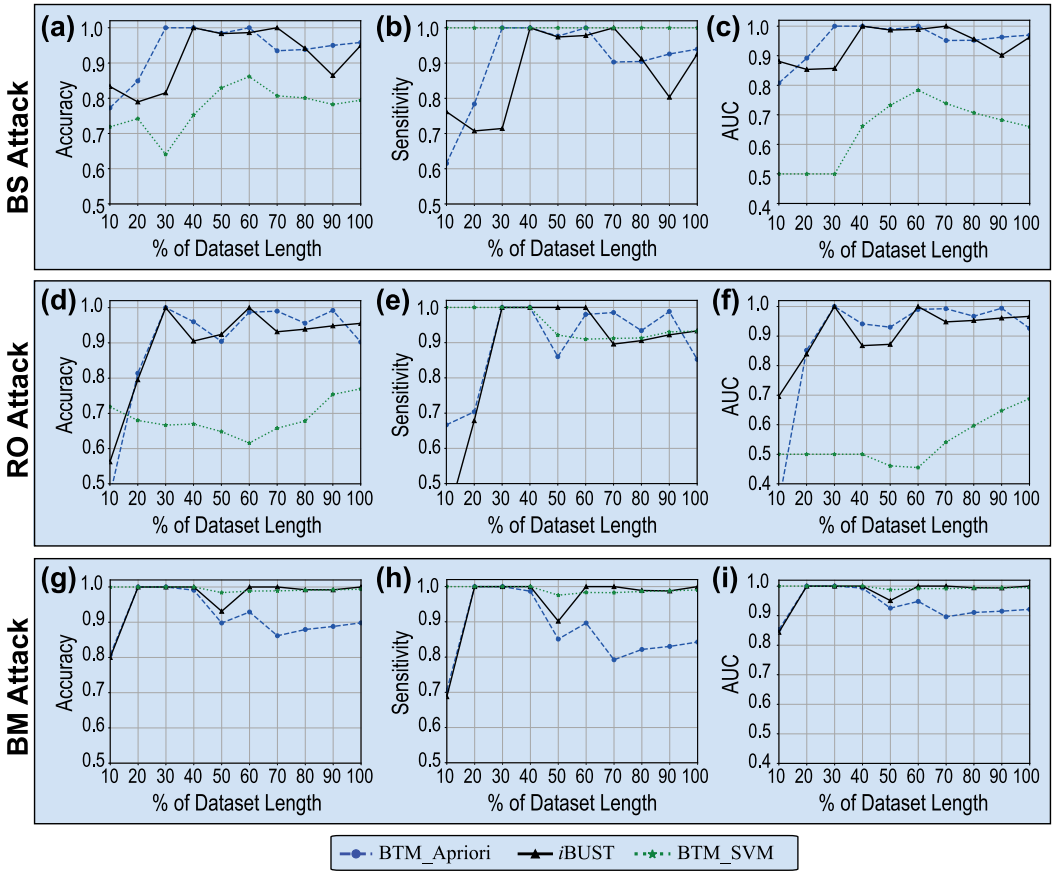

Fig. 11. Comparison of *i*BUST with other models for the TI attack against various indirect attacks.

signatures. Although, *i*BUST and BTM_Apriori exhibit comparable performance for RO attack; however, in case of average performance the proposed technique overpowers its ancestor for all three metrics.

5.3.2. Performance under CX attack

This attack is different from the CN attack since here Y performs untrustworthy activities for one or multiple context(s) while performing trustworthy activities for the rest of the contexts, and thus, maintains a good reputation to X. For instance, Y acts untrustworthy for "E-commerce" related activities in DIUD and acts trustworthy for the rest of the contexts. The acquired results of this attack are presented in Fig. 10a to 10i.

The results exhibit the dominance of *i*BUST over BTM_Apiori. As can be observed from the figures is that the performance of the proposed model is better for most of the instances over BTM_Apriori for any indirect attack. For instance, the average Ac receives by $\mathcal{E}$FP-Growth in the case of BS, RO, and BM are 0.938, 0.850, and 0.984, respectively in comparison to that of Apriori, which are 0.892, 0.845, and 0.955, respectively. This shows that the behavioural signatures generated by the $\mathcal{E}$FP-Growth algorithm are more accurate to identify context-aware attacks over the other two models. However, analogous to the CN attack, SVM shows relatively lower performance for both the indirect attacks for the same reasons that are mentioned in Section 5.3.1.

5.3.3. Performance under TI attack

In TI attack, a user, Y would try to gain a good reputation to X by alternating its behaviours between honest and dishonest after fixed intervals. The acquired results of this attack are presented in Fig. 11a to 11i.

It can be observed from the figures that the BTM_SVM model is severely affected by both the indirect attacks for Ac and AUC metrics in case of BS and RO attacks. Herein, for BS and RO, the SVM is unable to find the most representative hyperplane for separating the trust class due to altering the behaviour of the entities. However, it performs comparable to other models in case of BM attack for all 3 metrics.

Conversely, even through the BTM_Apriori suffers in case of BM attack, it demonstrates comparable results for other two attacks, namely BS and RO. The average Acs are: 0.903, 0.902, and 0.918 for BS, RO, and BM, respectively. On the other hand, both *i*BUST exhibits comparatively better performance than the other models for all 3 attacks. For instance, the average Acs of the proposed model are 0.903, 0.915, and 0.971 for BS, RO, and BM, respectively. A similar trend is also observed for other performance metrics, and hence, can be selected as the most suitable model among the compared models.

6. Discussion

In general, from the results of sensitivity for any attack (direct or indirect), it can be stated that BTM_SVM exhibits better performance, which implies that it can detect true positives more accurately than others. Again, between *i*BUST and BTM_Apriori, the former can detect true positives more accurately than the latter; hence, in most of the scenarios, the former shows better performance than the latter.

However, in BTM_SVM, the generated hyperplanes were unable to separate true negative entities in most of the scenarios, and hence, demonstrate relatively lower performance for Ac and AUC. Conversely, both association rule-based techniques demonstrate comparable performance in detecting true negative entities for the CN attack for any indirect attack. However, in case of a CX attack, BTM_Apriori fails to detect true positive and true negative entities as accurately as the proposed model, *i*BUST; and therefore, attains lower performance.

Based on the above analysis, it can be concluded that the proposed model, *i*BUST overpowers other compared models. Hence, it can be a

suitable alternative to the existing solutions for tackling various threats and attacks in the CPPS networks.

6.1. Limitations and future directions

One of the limitations of this work is that due to the unavailability (to the best of our knowledge) of the other relevant datasets for BTM, the *i*BUST was evaluated with only one dataset, i.e., DIUD. Hence, preparing a new dataset for BTM and analysing the performance of the *i*BUST with that dataset remains an important issue to be dealt in the future. Again, even though, the *i*BUST has embraced MEDF for adapting trust evolution; however, an extensive analysis is required to understand the trust evolution dynamics and propose adaptive models by capturing these dynamics for responding quickly.

7. Conclusion

In this paper, an intelligent BTM is proposed, named *i*BUST for securing the CPPS networks. The proposed model incorporates $\mathcal{E}$FP-Growth algorithm, which is a variant of the FP-Growth algorithm, where the latter is enhanced by altering the internal data structures for faster AR extraction time and by developing a modified exponential decay Function or MEDF for calculating minsupp automatically in order to facilitate the adaptation of trust evolution characteristics. In the proposed model, the trust classes are classified by employing the Naïve Bayes classifier. For evaluating the performance of *i*BUST with other existing models, a trust evolution-supported experimental environment is designed taking a benchmark dataset, called DIUD into consideration. According to the acquired results, the proposed model outperforms all its contenders and establishes its suitability among the compared models for the CPPS networks.

CRediT authorship contribution statement

Saiful Azad: Conceived the method and experiments, Implemented and conducted the experiment, Contributed to writing the paper both in the draft and final version. **Mufti Mahmud:** Conceived the method and experiments, Contributed to experiments and in analysing the results, Contributed to writing the paper both in the draft and final version. **Kamal Z. Zamli:** Analysed the results, Contributed to writing the paper both in the draft and final version. **M. Shamim Kaiser:** Analysed the results, Contributed to writing the paper both in the draft and final version. **Sobhana Jahan:** Analysed the results, Contributed to writing the paper both in the draft and final version. **Md. Abdur Razzaque:** Analysed the results, Contributed to writing the paper both in the draft and final version.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work has been supported in part by the FRGS project under Grant No. FRGS/1/2019/ICT04/UMP/02/1 and by the RDU project under Grant No. RDU182201-3.

Ethical approval

This article does not contain any studies with human participants or animals. Hence, ethical approval was not needed.

Informed consent

As this article does not contain any studies with human participants or animals, informed consent was not applicable.

References

Adiba, F. I., Islam, T., Kaiser, M. S., Mahmud, M., & Rahman, M. A. (2020). Effect of corpora on classification of fake news using naive Bayes classifier. *International Journal of Automation, Artificial Intelligence and Machine Learning*, *1*(1), 80–92, Number: 1.

Ahmed, S., Hossain, M., Nur, S. B., Shamim Kaiser, M., Mahmud, M., et al. (2022). Toward machine learning-based psychological assessment of autism spectrum disorders in school and community. In *Proc. TEHI* (pp. 139–149).

Ahmed, S., et al. (2021). Artificial intelligence and machine learning for ensuring security in smart cities. In *Data-driven mining, learning and analytics for secured smart cities* (pp. 23–47). Springer.

Akhund, N. U., et al. (2018). Adeptness: Alzheimer's disease patient management system using pervasive sensors-early prototype and preliminary results. In *Proc. brain inform.* (pp. 413–422).

Al Banna, M., Ghosh, T., Taher, K. A., Kaiser, M. S., Mahmud, M., et al. (2020). A monitoring system for patients of autism spectrum disorder using artificial intelligence. In *Proc. brain informatics* (pp. 251–262).

Al-Maolegi, M., & Arkok, B. (2014). An improved apriori algorithm for association rules. *IJNLC*, *3*(1), 21–29.

AlArjani, A., et al. (2022). Application of mathematical modeling in prediction of COVID-19 transmission dynamics. *Arabian Journal for Science and Engineering*, 1–24.

Asif-Ur-Rahman, M., Afsana, F., Mahmud, M., Kaiser, M. S., Ahmed, M. R., Kaiwartya, O., et al. (2019). Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things. *IEEE Internet of Things Journal*, *6*(3), 4049–4062.

Bhapkar, H. R., et al. (2021). Rough sets in COVID-19 to predict symptomatic cases. In *COVID-19: Prediction, decision-making, and its impacts* (pp. 57–68). Springer.

Bicaku, A., Maksuti, S., Palkovits-Rauter, S., Tauber, M., Matischek, R., Schmittner, C., et al. (2017). Towards trustworthy end-to-end communication in industry 4.0. In *Proc. INDIN* (pp. 889–896).

Biswas, M., Kaiser, M. S., Mahmud, M., Al Mamun, S., Hossain, M., Rahman, M. A., et al. (2021). An XAI based autism detection: The context behind the detection. In *Proc. brain informatics* (pp. 448–459).

Biswas, M., et al. (2021a). ACCU3RATE: A mobile health application rating scale based on user reviews. *PLoS One*, *16*(12), Article e0258050.

Biswas, M., et al. (2021b). Indoor navigation support system for patients with neurodegenerative diseases. In *Proc. brain inform* (pp. 411–422).

Blaze, M., Feigenbaum, J., Ioannidis, J., & Keromytis, A. (1999). The keynote trust-management system. In *Proc. int. workshop security protocols* (pp. 59–63).

Bllagdham, A. S. S., & Azad, S. (2020). Behavioral trust model. BTM 1.0. (Accessed 21 April 2020).

Borgelt, C. (2012). Frequent item set mining. *WIDM*, *2*(6), 437–456.

Chen, J., & Xu, G. (2009). Svm-based swift trust rating model in e-commerce. In *Proc. ETCS. Vol. 1* (pp. 640–643).

D'Angelo, G. (2019). Dishonest internet users dataset summary. (Accessed 10 May 2020).

D'Angelo, G., & Rampone, S. (2015). An artificial intelligence-based trust model for pervasive computing. In *10th International conference on P2P, parallel, grid, cloud and internet computing* (pp. 701–706). IEEE.

D'Angelo, G., Rampone, S., & Palmieri, F. (2017). Developing a trust model for pervasive computing based on apriori association rules learning and bayesian classification. *Soft Computing*, *21*(1), 6297–6315.

Das, S., Yasmin, M. R., Arefin, M., Taher, K. A., Uddin, M. N., & Rahman, M. A. (2021). Mixed bangla-english spoken digit classification using convolutional neural network. In M. Mahmud, M. S. Kaiser, N. Kasabov, K. Iftekharuddin, & N. Zhong (Eds.), *Applied intelligence and informatics, communications in computer and information science* (pp. 371–383). Cham: Springer International Publishing.

Davies, D. L., & Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*, 224–227.

Eyedali Mirjalili (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*(1), 120–133.

Farhin, F., Kaiser, M. S., & Mahmud, M. (2020). Towards secured service provisioning for the internet of healthcare things. In *Proc. AICT* (pp. 1–6).

Farhin, F., Kaiser, M. S., & Mahmud, M. (2021). Secured smart healthcare system: Blockchain and bayesian inference based approach. In *Proc. TCCE* (pp. 455–465).

Feng, W., Zhu, Q., Zhuang, J., & Yu, S. (2018). An expert recommendation algorithm based on pearson correlation coefficient and FP-Growth. *Cluster Computing*, *22*, 7401–7412.

Fournier-Viger, P. (2010). *Un modèle hybride pour le support à l'apprentissage dans les domaines procéduraux et mal-définis* (Ph.D. thesis), Du Doctorat En Informatique Cognitive, Montreal, Canada: University of Quebec in Montreal.

Fournier-Viger, P., Lin, J., Vo, B., Chi, T. T., Zhang, J., & Le, H. B. (2017). A survey of itemset mining. *WIDM*, *7*(4).

Gavriloaie, R., Nejdl, W., Olmedilla, D., Seamons, K. E., & Winslett, M. (2004). No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *Proc. Eur. semantic web symp.* (pp. 342–356). IEEE.

Ghosh, T., Al Banna, M. H., Rahman, M. S., Kaiser, M. S., Mahmud, M., Hosen, A. S., et al. (2021). Artificial intelligence and internet of things in screening and management of autism spectrum disorder. *Sustainable Cities and Society*, *74*, Article 103189.

Giandomenico, N., & de Groot, J. (2018). Insider vs. Outsider data security threats: What's the greater risk? In *Digital guardian's blog*.

Goethals, B. (2004). Frequent itemset mining dataset repository. (Accessed 10 May 2020).

Grønlund, A., Larsen, K. G., Mathiasen, A., & Nielsen, J. S. (2017). Fast exact k-means, k-medians and bregman divergence clustering in 1d. CoRR, abs/1701.07204.

Han, G., He, Y., Jiang, J., Wang, N., Guizani, M., & Ansere, J. A. (2019). A synergetic trust model based on svm in underwater acoustic sensor networks. *IEEE TVT*, *68*(11), 11239–11247.

International Telecommunication Union (2017). Overview of trust provisioning for information and communication technology infrastructures and services. (Accessed 22 June 2020).

Islam, N., et al. (2021). Towards machine learning based intrusion detection in iot networks. *Computers, Materials and Continua*, *69*(2), 1801–1821.

Jayasinghe, U., Lee, G. M., Um, T.-W., & Shi, Q. (2019). Machine learning based trust computational model for iot services. *IEEE TSUSC*, *4*(1), 39–52.

Jayasinghe, U., Truong, N. B., Lee, G. M., & Um, T.-W. (2016). RPR: A trust computation model for social internet of things. In *Proc. UIC* (pp. 930–937). IEEE.

Jesmin, S., Kaiser, M. S., & Mahmud, M. (2020). Artificial and internet of healthcare things based Alzheimer care during COVID 19. In *Proc. brain inform.* (pp. 263–274).

Kaiser, M. S., et al. (2021). 6 g access network for intelligent internet of healthcare things: Opportunity, challenges, and research directions. In *Proc. TCCE* (pp. 317–328).

Kumar, S., et al. (2021). Forecasting major impacts of COVID-19 pandemic on country-driven sectors: Challenges, lessons, and future roadmap. *Personal and Ubiquitous Computing*, 1–24.

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber–physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23.

Lin, Z., & Dong, L. (2017). Clarifying trust in social internet of things. *IEEE TKDE*, *30*(2), 234–248.

Lord, N. (2017). Insiders vs. outsiders: What's the greater cybersecurity threat? (infographic). In *Digital guardian's blog. Data insider*.

Mahmud, M., & Kaiser, M. S. (2021). Machine learning in fighting pandemics: A COVID-19 case study. In *COVID-19: Prediction, decision-making, and its impacts* (pp. 77–81). Springer.

Mahmud, M., Kaiser, M. S., Hussain, A., & Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2063–2079.

Mahmud, M., Kaiser, M. S., McGinnity, T. M., & Hussain, A. (2021). Deep learning in mining biological data. *Cognitive Computation*, *13*(1), 1–33.

Mahmud, M., Kaiser, M. S., & Rahman, M. A. (2022). Towards explainable and privacy-preserving artificial intelligence for personalisation in autism spectrum disorder. In M. Antona, & C. Stephanidis (Eds.), *Lecture notes in computer science*, *Universal access in human-computer interaction. User and context diversity* (pp. 356–370). Cham: Springer International Publishing.

Mahmud, M., Kaiser, M. S., Rahman, M. M., Rahman, M. A., Shabut, A., Al-Mamun, S., et al. (2018). A brain-inspired trust management model to assure security in a cloud based iot framework for neuroscience applications. *Cognitive Computation*, *10*(5), 864–873.

Mahmud, M., et al. (2019). A brain-inspired trust management model to assure security in a cloud based iot framework for neuroscience applications. *Cognitive Computation*, *10*(5), 864–873.

Mahmud, M., et al. (2022). Towards explainable and privacy-preserving artificial intelligence for personalisation in autism spectrum disorder. In *Proc. HCII* (pp. 356–370).

Monostori, L., Kádár, B., Bauernhans, T., Kondoh, S., Kumara, S., Reinhart, G., et al. (2016). Cyber-physical systems in manufacturing. *CIRP Annals*, *65*(2), 621–641.

Nahiduzzaman, M., et al. (2020). Machine learning based early fall detection for elderly people with neurological disorder using multimodal data fusion. In *Proc. brain inform.* (pp. 204–214).

Nasrin, F., Ahmed, N. I., & Rahman, M. A. (2021). Auditory attention state decoding for the quiet and hypothetical environment: A comparison between bLSTM and SVM. In M. S. Kaiser, A. Bandyopadhyay, M. Mahmud, & K. Ray (Eds.), *Advances in intelligent systems and computing*, *Proceedings of TCCE* (pp. 291–301). Singapore: Springer.

Nawar, A., Toma, N. T., Al Mamun, S., Kaiser, M. S., Mahmud, M., & Rahman, M. A. (2021). Cross-content recommendation between movie and book using machine learning. In *2021 IEEE 15th international conference on application of information and communication technologies* (pp. 1–6).

Nejdl, W., Olmedilla, D., & Winslett, M. (2004). Peertrust: Automated trust negotiation for peers on the semantic web. In *Proc. VLDB workshop on SDM* (pp. 118–132).

Noor, M. B. T., Zenia, N. Z., Kaiser, M. S., Mamun, S. A., & Mahmud, M. (2020). Application of deep learning in detecting neurological disorders from magnetic resonance images: A survey on the detection of Alzheimer's disease, Parkinson's disease and schizophrenia. *Brain Informatics*, *7*(1), 1–21.

Nourian, A., & Madnick, S. (2018). A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet. *IEEE TDSC*, *15*(1), 2–13.

Paul, A., et al. (2022). Inverted bell-curve-based ensemble of deep learning models for detection of COVID-19 from chest X-rays. *Neural Computing and Applications*, 1–15.

Pawlick, J., & Zhu, Q. (2017). Strategic trust in cloud-enabled cyber–physical systems with an application to glucose control. *IEEE TIFS*, *12*(12), 2906–2919.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-Learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prakash, N., et al. (2021). Deep transfer learning for COVID-19 detection and infection localization with superpixel based segmentation. *Sustainable Cities and Society*, *75*, Article 103252.

Rabby, G., Azad, S., Mahmud, M., Zamli, K. Z., & Rahman, M. M. (2020). Teket: A tree-based unsupervised keyphrase extraction technique. *Cognitive Computation*, *12*(4), 811–833.

Rabby, G., et al. (2018). A flexible keyphrase extraction technique for academic literature. *Procedia Computer Science*, *135*, 553–563.

Rahman, M. A. (2018). *Gaussian process in computational biology: Covariance functions for transcriptomics* (phd), University of Sheffield.

Rahman, M. A., Brown, D. J., Mahmud, M., Shopland, N., Haym, N., Sumich, A., et al. (2022). Biofeedback towards machine learning driven self-guided virtual reality exposure therapy based on arousal state detection from multimodal data. In *Proc. BI2022* (pp. 1–12).

Rahman, M. A., Brown, D. J., Shopland, N., Burton, A., & Mahmud, M. (2022). Explainable multimodal machine learning for engagement analysis by continuous performance test. In M. Antona, & C. Stephanidis (Eds.), *Lecture notes in computer science*, *Universal access in human-computer interaction. User and context diversity* (pp. 386–399). Cham: Springer International Publishing.

Rahman, M. A., Brown, D. J., Shopland, N., Harris, M. C., Turabee, Z. B., Heym, N., et al. (2022). Towards machine learning driven self-guided virtual reality exposure therapy based on arousal state detection from multimodal data. In M. Mahmud, J. He, S. Vassanelli, A. van Zundert, N. Zhong (Eds.), *Brain informatics* (pp. 195–209). Cham: Springer International Publishing.

Rakib, A. B., Rumky, E. A., Ashraf, A. J., Hillas, M. M., & Rahman, M. A. (2021). Mental healthcare chatbot using sequence-to-sequence learning and bilstm. In M. Mahmud, M. S. Kaiser, S. Vassanelli, Q. Dai, & N. Zhong (Eds.), *Brain informatics* (pp. 378–387). Cham: Springer International Publishing.

Sadik, R., Reza, M. L., Al Noman, A., Al Mamun, S., Kaiser, M. S., & Rahman, M. A. (2020). COVID-19 pandemic: A comparative prediction using machine learning. *International Journal of Automation, Artificial Intelligence and Machine Learning*, *1*(1), 1–16.

Sarkar, I. H. (2019). A machine learning based robust prediction model for real-life mobile phone data. *Internet Things*, *5*(1), 180–193.

Satu, M. S., et al. (2021). Short-term prediction of COVID-19 cases using machine learning models. *Applied Sciences*, *11*(9), 4266.

Srikant, R., & Agarwal, R. (1996). Mining quantitative association rules in large relational tables. In *Proc. of ACM SIGMOD* (pp. 1–12). ACM.

Sumi, A. I., et al. (2018). Fassert: A fuzzy assistive system for children with autism using internet of things. In *Proc. brain inform.* (pp. 403–412).

Van Rossum, G., & Drake, F. L., Jr. (1995). Python tutorial. In *Centrum voor Wiskunde en Informatica Amsterdam*. The Netherlands.

Veness, C. (2016). Calculate distance, bearing and more between latitude/longitude points. Movable Type Scripts.

Wadhera, T., & Mahmud, M. (2022a). Brain networks in autism spectrum disorder, epilepsy and their relationship: A machine learning approach. In *Artificial intelligence in healthcare: Recent applications and developments* (pp. 125–142). Springer.

Wadhera, T., & Mahmud, M. (2022b). Computing hierarchical complexity of the brain from electroencephalogram signals: a graph convolutional network-based approach. In *Proc. IJCNN* (pp. 1–6).

Wadhera, T., & Mahmud, M. (2022c). Influences of social learning in individual perception and decision making in people with autism: A computational approach. In *Proc. brain inform.* (pp. 50–61).

Wadhera, T., & Mahmud, M. (2023). Brain functional network topology in autism spectrum disorder: A novel weighted hierarchical complexity metric for electroencephalogram. *IEEE Journal of Biomedical and Health Informatics*, *27*(4), 1718–1725.

Wang, Y. (2018). Trust quantification for networked cyber–physical systems. *IEEE Internet of Things Journal*, *5*(3), 2055–2070.

Wang, Z., & Lin, Z. (2019). Optimal feature selection for learning-based algorithms for sentiment classification. *Cognitive Computation*, *12*, 238–248.

Wang, C., & Zheng, X. (2020). Application of improved time series apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint. *Evolutionary Intelligence*, *13*, 39–49.

Xiong, L., & Liu, L. (2003). A reputation-based trust model for peer-to-peer e-commerce communities. In *Proc. IEEE CEC* (pp. 275–284).

Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for internet of things. *Journal of Network and Computer Applications*, *42*, 120–134.

Zaman, S., et al. (2021). Security threats and artificial intelligence based counter-measures for internet of things networks: A comprehensive survey. *IEEE Access*, *9*, 94668–94690.

Zamli, K. Z., Din, F., Nasser, A. B., & Alsewari, A. (2020). Combinatorial test suite generation strategy using enhanced sine cosine algorithm. In *Proc. of ECCE* (pp. 127–137).

Zhang, Z.-X. (2001). The effects of frequency of social interaction and relationship closeness on reward allocation. *The Journal of Psychology*, *135*(2), 154–164.

Zhang, Y., Chen, H., & Wu, Z. (2006). A social network-based trust model for the semantic web. In *Proc. ATC* (pp. 183–192).

Zhang, J., Li, Y., Xiao, W., & Zhang, Z. (2023). Online spatiotemporal modeling for robust and lightweight device-free localization in nonstationary environments. *IEEE Transactions on Industrial Informatics*, *19*(7), 8528–8538.

Zhang, J., Zhao, Y., Shone, F., Li, Z., Frangi, A. F., Xie, S. Q., et al. (2022). Physics-informed deep learning for musculoskeletal modeling: Predicting muscle forces and joint kinematics from surface EMG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*, 484–493.

Zhao, H., Sun, D., Yue, H., Zhao, M., & Cheng, S. (2018). Dynamic trust model for vehicular cyber–physical systems. *IJNS*, *20*(1), 157–167.