IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 31, 2017, accepted March 2, 2017, date of publication March 21, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2683159

Computational Offloading for Efficient Trust Management in Pervasive Online Social Networks Using Osmotic Computing

VISHAL SHARMA[1], ILSUN YOU [1], (Senior Member, IEEE), RAVINDER KUMAR[2], AND PANKOO KIM[3]

[1]Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, South Korea
[2]Computer Science and Engineering Department, Thapar University, Patiala 147004, India
[3]Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea

Corresponding author: I. You (ilsunu@gmail.com)

This work was supported in part by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea under Grant NRF-2014H1C1A1073115 and in part by the Soonchunhyang University Research Fund.
This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author.

ABSTRACT Pervasive social networking (PSN) aims at bridging the gap between the services and users by providing a platform for social communication irrespective of the time and location. With the advent of a new era of high-speed telecommunication services, mobile users have evolved to a large extent demanding secure, private, and trustworthy services. Online social networks have evolved as pervasive online social networks (POSNs), which uses a common platform to connect users from hybrid applications. Trust has always been a concern for these networks. However, existing approaches tend to provide application-specific trust management, thus resulting in the cost of excessive network resource utilization and high computations. In this paper, a pervasive trust management framework is presented for POSNs, which is capable of generating high trust value between the users with a lower cost of monitoring. The proposed approach uses a flexible mixture model to develop the system around six different properties, and then utilizes the concept of osmotic computing to perform computational offloading, which reduces the number of computations as well as computational time. The novel concepts of lock door policy and intermediate state management procedure are used to allow trust visualization by providing efficient identification of trustworthy and untrustworthy users. The proposed approach is capable of predicting user ratings efficiently with extremely low errors, which are in the range of ±2%. The effectiveness of the proposed approach is demonstrated using theoretical and numerical analyses along with data set-based simulations.

INDEX TERMS Pervasive social networks, online social networks, trust management, osmotic computing, trust visualization.

I. INTRODUCTION

With an increase in the number of users across the social communication and mobile platforms, Pervasive Social Networking (PSN) has evolved to a great extent. PSN aims at providing a platform for connectivity to all without considering the geographical or application barriers for different users [1]. Nowadays, the growth of next generation of mobile systems leads to PSN providing various efficient applications such as chat services, recommender systems, decision support systems, and gaming modules to users without compromising the connectivity between them [2], [3].

Another key domain of PSN is the Pervasive Online Social Networks (POSNs). POSNs similar to other social platforms use common connectivity platform which can ensure communication at any time and in any location. POSNs are composed of different online social networking platforms which bring together different users to share data and services efficiently and securely.

With most of the users being mobile and having high dynamics, trust is always a key concern for POSNs. Although POSNs enhance the social communication applications and experiences of every online user, yet these suffer from a major issue of trust management [4]. Trust can be confirmed by a user, service provider or a simple connection between any two entities of the network. Efficient trust formation allows enhanced device management with the maintenance of privacy and confidentiality of users as well as data [5], [6]. With applications ranging from data storage to high-end gaming, trust plays a key role in maintaining connectivity and authentication between the users [7], [8]. Anonymity and authentication can be improved by provisioning of efficient trust mechanisms [9].

Over the past few years, trust management has been a concern for different types of networks, such as social networks, ad hoc networks, POSNs, and behavioral analyses networks. There are a large number of approaches available, such as anonymous authentication by Yan *et al.* [10], social behavior analyses by Zhang *et al.* [11], privacy preserving for video surveillance by Carniani *et al.* [12], and multi-dimensional trust management by Yan *et al.* [13], which provide a variety of solutions for resolving trust issues depending on the types of applications. However, computational offloading, cost of monitoring, and relation between users are ignored by these approaches making them suitable only for a particular type of network conditions.

A. MOTIVATION AND PROBLEM STATEMENT

POSNs aim at connecting all without any barrier of time and location in a flexible and ubiquitous manner. With new technologies revolutionizing the current era of telecommunication services, POSNs provide support to multiple users with heterogeneous connectivity. With a large number of users interacting with different operators, social service providers, and application gateways, it is extremely important to provide a trustworthy environment. The trustworthy environment provides reliable, secure, and personalized connectivity between information seekers and information providers. Trust management allows exhaustive utilization of network services by diversified users with mutual consent over social behaviors especially focusing on user privacy and confidentiality. Thus, due to the impact of POSNs, it becomes important to consider trust-management and evaluation for efficient social communications.

It is of paramount significance to manage trust as well as define policies for it. However, trust management is not only the requirement for POSNs. Trust management, evaluation, and policy formulation rely heavily on fast and distributed computations. There are considerable overheads related to the evaluation of a large number of users operating in different connectivity environment which makes it difficult for handling a large set of trust policies. Thus, the cost of monitoring is a massive issue for POSNs. Further, the decision on computational offloading and divisibility of service operations are other issues to be resolved for provisioning of efficient trust management with a lower cost of monitoring.

B. OUR CONTRIBUTION AND KEY HIGHLIGHTS

In this paper, a novel solution for trust management in POSNs is proposed. The proposed approach presents a pervasive trust management framework which uses the concept of relation cost that operates over the Flexible Mixture Model (FMM) [14]. The proposed solution uses the concept of learning and pre-hand prediction of users' trust. The proposed approach primarily focuses on lowering the cost of monitoring along with the identification of trustworthy and untrustworthy users.

In order to overcome the computational overheads involved in the trust management over POSNs, the concept of osmotic computing, which is proposed as a new paradigm for edge computing by Villari *et al.* [15], is applied over social communications. An efficient computational offloading mechanism is adopted to lower the cost of monitoring. Further, the concepts of lock door policies and intermediate state management procedure are proposed to efficiently manage the trust policies for users, servers, and source-connections. The movements of application specific data and control over osmotic environment for POSNs are performed by using three different approaches, namely, bio-inspired movement by using Ant Colony Optimization (ACO) [16] and Artificial Bee Colony Optimization (ABC) [17], probabilistic movement, and threshold-based movement. Further, the n-polygon solution is demonstrated for trust visualization which allows efficient identification of trustworthy as well as untrustworthy users in POSNs. The other key highlights of the proposed approach are:

- Formation of an intelligent trust management solution along with trust visualization.
- Efficient computational offloading using the concept of osmotic computing.
- Lower cost of monitoring and osmosis time for handling a large number of users in POSNs.

Rest of the paper is structured as follows: Section II gives details of existing literature. Section III gives an overview of background and formulation of the system model. Section IV presents a detailed proposed pervasive trust management framework. Section V gives theoretical and numerical analyses of the proposed solution. Section VI demonstrates the efficiency of the proposed solution using simulation dataset. Section VII gives state-of-the-art comparison and discussions along with open issues. Finally, Section VIII concludes the paper.

II. RELATED WORK

PSN has seen a tremendous growth over the past few years. With an aim at establishing connectivity to all irrespective to the classification of users, PSN relies heavily on the management of trust between the entities [2]. With the advent of a large number of online social networks, connectivity to all paradigms has revolutionized the formation of POSNs. Handling a large number of users, and allowing mutual coordination and peering between them has been the primary objective of POSNs. Security and privacy management are the other key issues in POSNs [7], [18].

A. REPUTATION AND BEHAVIOR ANALYSES SYSTEMS

Behavioral analyses systems use node patterns to manage the reliability of the network. Zhang *et al.* [11] developed a social behavior analyses system for PSNs. The authors presented a detailed study on the behavioral analyses along with the development of a pattern-based deep reinforcement-based learning system with its case study on different models.

Although efficient, yet the solution given by the authors have limited scope for being used in POSNs as user pattern may vary with time and real-time evaluations do not allow efficient pattern monitoring.

Machado *et al.* [19] focused on the pervasive data forwarding in mobile social networks. The authors utilized a real scenario and considered geographical properties for selecting forwarding path in the opportunistic network formed in the mobile social networks. The evaluation and specific implementation do not allow this approach to be used for POSNs.

Reputation refers to confidence generated for the nodes in a network by the other nodes operating at the same time. Content reputation and node trust can also be handled simultaneously, which can help efficient provisioning of user decisions [20]. Yan *et al.* [21] developed a practical system for reputation-based chat rooms. However, the proposed system has a limited scalability with implementation only in an ad hoc environment.

Sharma *et al.* [22] considered reputation and behavior management between the users which operate in the ad hoc environment. The node trust and behavior is calculated on the basis of coordination between ad hoc nodes. Their approach is not a centralized solution and can be computationally difficult to operate in POSNs.

B. TRUST-MANAGEMENT SYSTEMS

Trust management systems maintain a secure and reliable connectivity between the users in the cyber space [4]. Chen *et al.* [23] proposed a trust management system for IoT systems by considering less information management over capacity-limited nodes. The solution given by the authors is a service-based approach that may suffer from design issues. Although service systems are highly scalable, yet maintenance of node-trust is tedious with dependency only on software security.

Yan *et al.* [13], [24] considered ad hoc networks as a platform for PSN and developed a model for trust management between the nodes by using a key concept of attribute-based encryption. The proposed approach is secure in terms of data confidentiality but does not guarantee a generic implementation in all pervasive scenarios.

Short Message Services (SMS) are also a type of pervasive systems which need attention for efficient trust management between the senders and receivers. Chen *et al.* [23] focused on the developed of a robust and reliable system, which is capable of maintaining trust between the users of SMS. The issue with this approach is its entire dependency on the path between the source and the destination. Applicability to generic POSNs is still an issue with this approach.

Ma and Yan [25] developed a PSN controller which is capable of providing trust between the social network users. Accuracy, efficiency and robust trust management are the key advantages of this approach. Multi-platform support and scalability are yet to be evaluated for this approach preventing its use in POSNs.

PSN has evolved a lot over the past few years, but POSNs are still newer concepts in pervasive networking. POSNs incorporate the features of online social networks with an aim of providing trustworthy connectivity to all users irrespective of the domain and platform of connectivity. It is evident that existing approaches show potential to be used in POSNs, but these require a vast modification for actual implementation. Thus, a novel solution is required which can categorize the working of POSNs and can provide efficient trust management, prediction, and visualization for both trustworthy as well as untrustworthy users.

III. BACKGROUND AND SYSTEM MODEL

Maintaining trust in POSNs is affected by the number of users and the types of properties over which the trust is defined. Trust can be analyzed as a measure of secure connectivity between the users and the servers. In the considered system, trust is defined in terms of relation cost (R_c). The higher value for relation cost means a greater trust between the entities of POSNs. This section defines the properties and user model used to formulate the relation cost of all the users. Further, this section also presents a detailed mathematical modeling of the osmotic system which forms the backbone of the proposed framework.

A. USER MODEL AND PROPERTIES

POSNs are composed of a large number of users connecting without any geographical boundaries and time considerations. FMM is used for user modeling, which allows the formation of a probabilistic system to measure the relation cost of the network given any number of users and connections. A mixture model is used to define the system as it can formulate a network which is composed of a large number of users that cannot account for a single or constant probability in POSNs because of diversification in the type and number of connections. Further, unavailability of the community classification also supports the use of mixture model. However, in the scenarios where the subgroups are known for each user, a composition model can be applied which is bounded by constant values [14], [26]. Various properties used in the proposed approach are:

- Degree of connectivity (D_c): It defines the sum of in-degree and out-degree for every user in POSNs. A higher value denotes more control over the network as well as easy access to most of the network components and information.
- Depth of connectivity (D_e): It defines the reach of a particular user in POSNs. It denotes the connectivity of users to the farthest most users identified on the basis of labels. A user with a high degree of connectivity usually has a high depth of connectivity.
- Level of osmotic shifts (L_o): It defines the number of times a user is shifted to the osmotic servers from the actual cloud. In the proposed approach, the osmotic servers maintain the trust over the network. A detailed explanation is provided in the sections to follow.

- Trust violations (T_p): It defines the threat score associated with every user. It is calculated as a percentage of the number of times a user is a trust violator to the total violations occurred in the network.
- Computational cycles (C_v): It denotes the CPU cycles consumed by a user or server process.
- Memory Utilization (V_m): It denotes the memory consumed by a user or server process.
- Cost of monitoring (M_c): It defines the computational cost associated with the management of trust in POSNs. The cost of monitoring is calculated as the energy and memory consumed in managing and computing trust over the entire system.
- Computational Overheads (C_o): It denotes the latency and excessive iterations a system undergoes while performing computations for evaluation of relation cost of every user.

The entire system is modeled on the above given first six properties which form the set K such that $K = \{k_1, k_2, \ldots k_j\}$, where $j = 6$ and S is the set of different classes defined on the basis of dominance in properties such that $S = \{s_1, s_2, \ldots s_i\}$, where $6 \leq i \leq N$. Considering the number of properties initially defined, there can be a minimum of six different classes to which N users can belong. However, the division of users into classes can be customarily controlled depending on the number of categories into which users are to be presented. The equal number of classes and properties allows management of users directly depending on the dominance. For generalization, the number of classes can be set as $\frac{P}{b} \times j$, where b is the number of sub-groups formed for each property.

The classification helps in identifying the users which are to be considered for continuous monitoring. Let G_m be the membership of a user m in the set S and G_z be the membership of a property z in the set K such that $P(G_m)$ and $P(G_z)$ are the multinomial distribution [27] on the properties and user classes, respectively, such that

$$P(G_m) = \frac{|S|}{\prod_{m=1}^{|S|} m!} \prod_{m=1}^{|S|} G_m^m \tag{1}$$

and

$$P(G_z) = \frac{|K|}{\prod_{z=1}^{|K|} z!} \prod_{z=1}^{|K|} G_z^z. \tag{2}$$

G_m is calculated as the ratio of the number of sub-classes to which a user m belongs to the total user classes available in POSNs; and G_z is calculated as the ratio of users with property z to the total number of users in the POSNs. Now, considering D_p as the depending rating over the user class and properties such that D_p is a multinomial distribution over G_m and G_z given as $P(D_p|G_m, G_z)$. Thus, the relation cost for a user m is given as a joint probability over FMM, i.e. $R_c = P(m, z, D_p)$, which implies

$$R_c = \sum_{G_z, G_m} \mathbb{P}, \tag{3}$$

and

$$\mathbb{P} = P(G_m)P(G_z)P(m|G_m)P(m|G_z)P(D_p|G_z, G_m) \tag{4}$$

Now, from the concepts of FMM, training and prediction form the key part of relation cost calculations. With every iteration and variation in the property of a user, the value of R_c changes. From the definition of FMM [26], training of the system is controlled by a variable termed as clustering constant c such that the training relation cost is given as:

$$P(G_z, G_z|m, z, D_p) = \frac{\mathbb{P}^c}{\sum_{G_z, G_m} \mathbb{P}^c} \tag{5}$$

Since the considered system is an application over real-time instances, thus, the existing FMM prediction model cannot be readily applied as it depends on the posterior state for finalizing the user ratings. Hence, the prediction of relation cost R_c^p in the considered system is calculated using the entropy modeling [28] over each user w.r.t. to its R_c distribution across the entire social network. The prediction of the next value for a user is given as the deviation of its entropy at the current state R_c^e from the mean entropy $\overline{R_c^e}$ of the social network, such that

$$R_c^p = \sqrt{\frac{1}{x} \sum_{i=1}^{x} \left(R_{c,i}^e - \overline{R_c^e}\right)^2}. \tag{6}$$

Here

$$R_c^e = -\sum_{i=1}^{D_e} R_{c,i} \log\left(R_{c,i}\right), \tag{7}$$

and

$$\overline{R_c^e} = \frac{1}{N} \sum_{i=1}^{N} (R_c^e)_i, \tag{8}$$

where x is the previous iterations for which the user entropy is available. The predicted value can help in determining the future trend of the social network that allows the determination of users which may violate the trust properties.

B. OSMOTIC MODEL

Osmotic model is derived using the concept of osmotic computing, which is based on the chemical process of osmosis [15]. The process aims at balancing the concentration of the solution to provide a state of equilibrium on the either side of the semipermeable membrane. In the social model considered for evaluation in pervasive environment, the success of osmotic model depends on the appropriate selection of components, which include

- *Selection of Solute:* The solute forms the static part of the solution which cannot be moved across the membrane. In the considered social network, number of servers (V_s), energy of the servers (E_s), computational support (C_v), available memory (V_m), D_c, and D_e form the solute part.
- *Selection of Solvent:* Usually, osmosis aims at the transaction of services between the servers to allow smaller

services to be handled by near-user servers whereas high-end services are handled by large public/private edge cloud systems. However, this paper aims at the resolution of computational load of POSNs in managing trust, which can be attained by selection of a set of users that may violate the trust properties of the social networks. Thus, the number of users in the social networks is considered as the solvent for the osmotic model. Balancing the number of users and accurately shortlisting them for trust management is the key objective of the proposed system.

- *Selection of Semipermeable membrane:* Movement in the solution is managed by the semipermeable membrane which allows the solvent to move across the entire solution so as to balance the net concentration of the model. For osmotic computing, the semipermeable membrane has to be an intelligent application which can consider the current network state, and can take a decision on moving the users to available servers. The semipermeable membrane is the Decision Support System (DSS) for osmotic computing. The positioning of semipermeable membrane and selection criteria for movement are the key issues to be resolved while implementing osmotic computing.
- *Concentration properties for Osmotic Model:* The concentration properties define the equilibrium of a solution. These manage the flow of users across the social network servers. The concentration of users across the social servers can be modeled on the basis of M_c and C_o using [29].
  - M_c is the first property for osmosis which is modeled in terms of energy consumed per computation considering the CPU cycles, memory utilization, and services across the servers. The actual cost of monitoring is divided into three parts, namely, per user cost of monitoring, per server cost of monitoring, and overall network cost of monitoring, defined at a particular instance t such that:

$$M_{c,user} = D_c \sum_{i=1}^{r} (E_{s,server})_i, \tag{9}$$

    where

$$E_{s,server} = \frac{C_{v,server} \times V_{m,server}}{V_{m,s}} \times t. \tag{10}$$

    Here, r is the number of servers accessed by a user ($\max(r) = V_s$), $E_{s,server}$ is the energy consumed over the utilized server, $C_{v,server}$ is the computational cycle over utilized servers, $V_{m,server}$ is the memory used per utilized server and $V_{m,s}$ is the available memory resources over utilized server. Per server cost of monitoring is given as:

$$M_{c,server} = \left(1 - \frac{N_h}{N}\right) \sum_{i=1}^{N_h} (E_{s,user})_i, \quad N_h < N, \tag{11}$$

    where

$$E_{s,user} = \frac{C_{v,user} \times V_{m,user}}{V_{m,s}} \times t, \tag{12}$$

    N_h is the number of users handled by a single server, $C_{v,user}$ is the number of computations over single user, and $V_{m,user}$ is the memory used by a particular user.
  - C_o is the second property to be evaluated for osmosis procedures using the available count of users. C_o is calculated as the time lapse between the submission of first step for calculating R_c and the generation of output. Also, it adds up with the time consumed in number of calculations performed per server during trust management. The network cost of monitoring is evaluated as:

$$M_{c,network} = \sum_{j=1}^{N} (D_e)_j \cdot \frac{N}{\sum_{i=1}^{V_s} (N_h)_i}. \tag{13}$$

During entire session of trust management across POSNs, a state of equilibrium should be maintained throughout the resources. This allows efficient computational offloading to handle a large number of users as well as to detect the users which violate the trust properties. The detailed procedures on trust calculation and osmotic-based trust computational offloading are explained in the next section.

IV. PROPOSED APPROACH: PERVASIVE TRUST MANAGEMENT FRAMEWORK

The problem considered in this paper aims at managing trust across the social network users and servers without yielding high cost of monitoring. The major task of the proposed approach is to form an intelligent solution which not only provides a stabilized and adaptive solution for trust-enhancement, but can be used to efficiently handle the dynamics involved in the POSNs. The dynamics include sudden demand for scalability, prediction and estimation of trust state, intelligent decision making, post analyses of the network state, intermediate state management, updating the trust policies, and trust visualization.

The proposed approach forms the dynamic network which can handle a large number of users and can monitor them easily by shifting the marked users to other servers which are not the actual service providers but act as the watchdog for trust violators. The marking of users is done on the basis of trust policies. Visualization helps in presenting the results for efficient trust evaluations and management without excessive overheads.

In the proposed approach, trust is provided in POSNs using the concept of osmotic computing. Osmotic computing provides solution for dividing the services into sub-types which can be handled by the servers other than the hosting servers. Osmotic computing allows managing the users by shifting the critical users to the osmotic manager which shifts them to the appropriate osmotic server for monitoring and

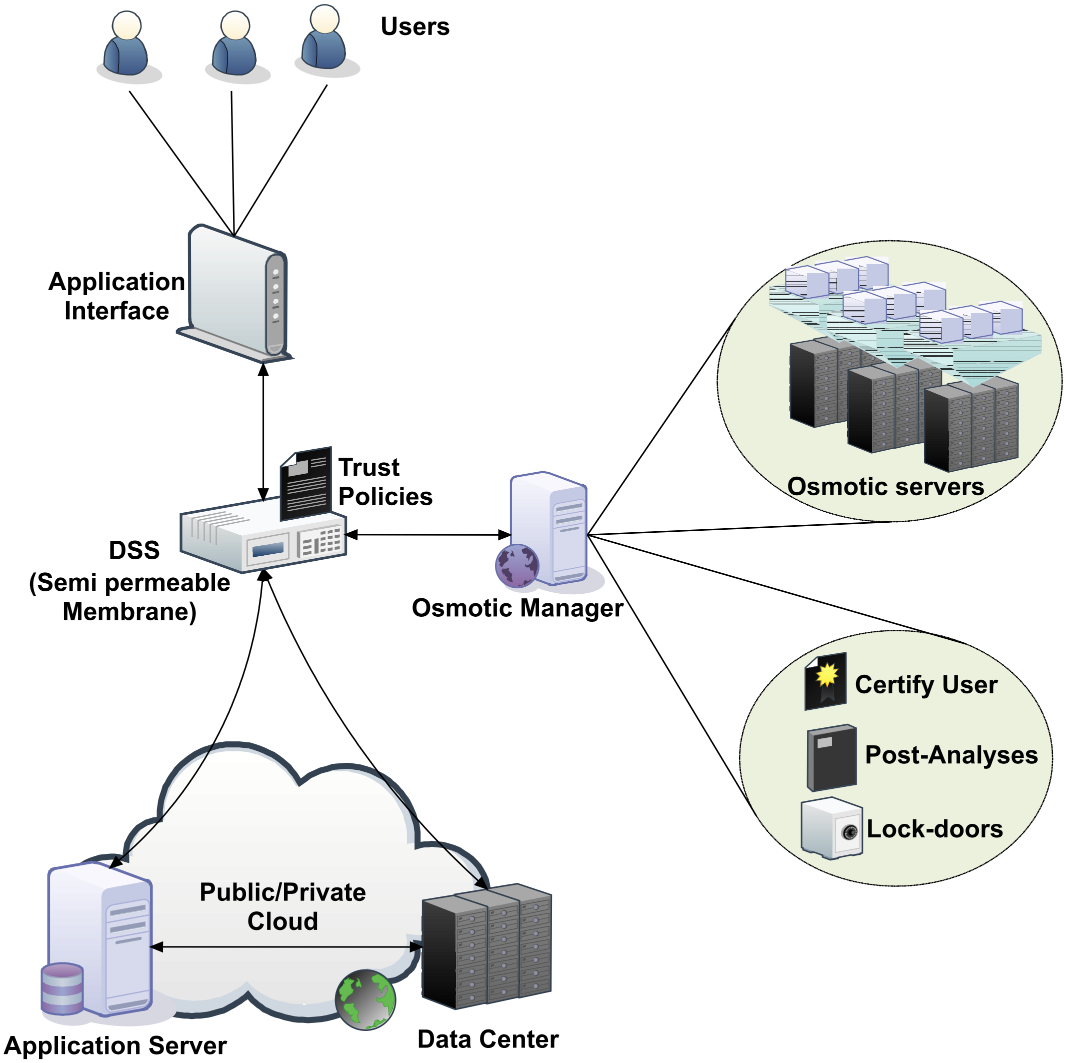


FIGURE 1. An illustration of pervasive trust management framework using osmotic computing.

calculating their trust on each interaction. An illustration of the proposed pervasive trust management framework using osmotic computing is shown in Fig. 1.

The framework categorizes the users on the basis of their activity by using the trust policies, and then continues to monitor them without interrupting the normal operations of the network. The framework comprises a DSS which forms the semipermeable membrane and contains the trust policies. It decides on maintaining the concentration of users across the entire social network. The DSS passes all the queries directly to the public/private cloud system, and at the same time interacts with the edge cloud system which is the near user osmotic system via osmotic manager. The osmotic manager further acts as a semipermeable membrane for the users shifted to the osmotic layer in order to maintain the concentration of users across the osmotic server. The osmotic server provides user trust values to the osmotic manager, which forms the visualization set and transfers them to DSS. The DSS transfers the visualized maps to core service provider for taking a decision on allowing a user or not.

The main applications of the osmotic manager include certifying a user, operating lock-door policies to monitor the excessive activity of users, and the post-analyses which otherwise would leverage excessive overheads on the actual servers and data centers. The simple shifting of services by dividing them on the basis of the number of users reduces the operational time of every server making the entire process to operate with lower overheads. The reduction in the number of operations over each server allows efficient management as well as form the base for the intermediate state management protocol.

A. TRUST POLICIES AND OPTIMIZATION PROBLEM

The trust policies consider three major role players, namely, user trust, server trust, and connection trust. Trust policies are implemented as optimization issues over the entire network.

The violation of any optimization criteria makes a user, server or connection vulnerable as well as untrustworthy.

- User Trust: User trust is the key component of the proposed system. It allows considering each user for excessive monitoring over the osmotic servers. Every user passes its requests directly to the application server. However, a DSS dedicatedly operates over the user requests and fetch properties of a user from the application server, which are used to derive the trust value in the form of relation cost. Violation of trust policies makes DSS shift monitoring of users to the osmotic manager that keeps on recording the activity of a user without disclosing its monitoring policies. The user trust is formed on R_c, $M_{c,user}$, and R_c^e. Every user should abide by the rules of trust which are defined below:
  - The maximum permissible variation in R_c for a user at any instance from its previous state should not be more than the deviation of its previous values from the mean of total network relation cost, i.e., for a user y,

$$(R_c(y))_t - (R_c(y))_{t-1} \leq \sqrt{\frac{1}{x}\sum_{i=1}^{x}\left(R_{c,i}(y) - \overline{R_c}\right)^2}. \tag{14}$$

  - The deviation in the cost of monitoring for a user should not be greater than the mean cost of entire network, i.e.,

$$\sqrt{\frac{1}{x}\sum_{i=1}^{x}\left(X_i - \overline{X}\right)^2} \leq \frac{1}{N}\sum_{i=1}^{N}(M_{c,user})_i, \tag{15}$$

where

$$X = M_{c,user}.$$

  - The current observed value for user entropy should not be greater than the predicted entropy, i.e.

$$R_c^e(y)_t \leq R_c^p(y)_t. \tag{16}$$

- Server Trust: Server trust is the secondary evaluation criteria which are invoked when the osmotic manager is unable to distinguish the trustworthy users from untrustworthy users. The variation in server trust allows considering all the users operating on a particular server to be fair or unfair depending on its current trust value. The server trust is evaluated using the cost of monitoring such that the deviation in the cost of monitoring for a single server should not be greater than the mean cost of all the servers available over the network, i.e.,

$$\sqrt{\frac{1}{x}\sum_{i=1}^{x}\left(Y_i - \overline{Y}\right)^2} \leq \frac{1}{V_s}\sum_{i=1}^{V_s}(M_{c,server})_i, \tag{17}$$

where

$$Y = M_{c,server}.$$

Entropy trust and relation cost trust can also be considered for evaluating the server trust conditions. However, the primary target of the proposed approach is to manage the user trust without overloading the servers with the burden of excessive computations. Thus, server trust is evaluated only over the cost of monitoring.

- Connection Trust: Apart from user trust and server trust, a large number of connections are made between the users and the servers despite the variations in the cost of monitoring and other properties defined above. Each connection is subjected to a unique trust value by which two entities in the POSNs ensures faith for efficient communications. The connection trust can be defined as the difference in the relation cost of two entities which request a connection. A connection between the entities is defined over R_c since it is a probabilistic value that allows easy mapping between them. Connection-trust can be defined as the similarity distance between the two probabilities each representing the relation cost of two connecting entities [30]. A condition on the similarity distance allows identification of trust over every connection made in the POSNs. For connection trust, the similarity distance between the R_c of two entities should not be greater than the similarity distance between their mean R_c, i.e.

$$D_{u_1,u_2} \leq D'_{u_1,u_2}, \tag{18}$$

where

$$D_{u_1,u_2} = \sqrt{(R_{c,u_1})^2 + (R_{c,u_2})^2}, \tag{19}$$

and

$$D'_{u_1,u_2} = \sqrt{(\overline{R_{c,u_1}})^2 + (\overline{R_{c,u_2}})^2}. \tag{20}$$

Here, u_1 and the u_2 are the two entities making connection with each other.

Considering the above defined trust policies, following optimization problems are formulated:

$$\begin{aligned} &\min\left(D_{u_1,u_2}\right),\\ &\min\left(M_{c,user}\right),\\ &\min\left(M_{c,server}\right),\\ &\max\left(R_c\right). \end{aligned} \tag{21}$$

B. USER MOVEMENT POLICIES FOR OSMOSIS

Osmotic computing allows evaluation of users which violate the trust policies to form a stable and consistent network. All the procedures considering the evaluations over the user properties are carried without many overheads and burden over the single server which hosts the application platform as well as interact with the data centers. Distribution of services has always been there in the form of load balancing, which includes shifting services across the servers that are connected to each other. However, there is always a concern of overheads and excessive computations that are induced when all the services are handling by a single layer of servers.

In this paper, users are to be shifted for the purpose of consistent monitoring and management of trust across the POSNs. All users which violate the rules defined as the trust policies are to be monitored until they start obeying the network policies. The osmotic manager receives all the users that violate the properties from the DSS, and then takes a decision on sending users to different servers. The osmotic manager can move the users by different ways. In this paper, three different ways are identified for moving users across the servers, which are:

1) FITNESS-BASED MOVEMENT

Fitness-based movement is derived by the optimization over a fitness function which controls the movement of users across the servers. Fitness function can be derived in number of ways depending on the complexity of model considered for POSNs. In this paper, ACO [16] and ABC [17], [31] are used to shift the users amongst the servers on the osmotic layer. The fitness objectives are determined on the basis of the dominance of a particular parameter. In the proposed approach, M_c is treated as the dominant parameter. Hence, the maximum or minimum value of fitness function is derived w.r.t. M_c.

- ACO-based osmosis: ACO is performed over the deposit of pheromone by the ants. Considering the similar property, ACO-based osmosis of performed by selecting the quantity of pheromone present over the osmotic manager for available osmotic servers. The pheromone is the server trust which allows selecting a server which can sustain more users in comparison with the other servers. Every user is treated as an ant and on the basis of trust, these are shifted to the osmotic manager, which takes into consideration the pheromone impact of users over the servers. The pheromone-based fitness function for user mobility $Move_u$ is given as [16]:

$$Move_u = \max\left(\frac{M_{c,user}^{\eta_1} R_c^{\eta_2}}{\sum_{i=1}^{x} M_{c,user}^{\eta_1} R_c^{\eta_2}}\right), \tag{22}$$

  where η_1 and η_2 are the balancing constants [16] such that $R_c^e \geq \eta_1$; and η_2 is selected such that $\eta_2 \geq 1$ and $\eta_2 \geq \eta_1$. The user with a maximum value for $Move_u$ is moved first to the osmotic manager for monitoring by osmotic server until its value is extremely lowered for pheromone deposit. Contrary to this, the pheromone-based fitness function for selection of server $Move_s$ is given as [16]:

$$Move_s = \min\left(\frac{M_{c,server}^{\eta_3} R_c^{\eta_4}}{\sum_{i=1}^{x} M_{c,server}^{\eta_3} R_c^{\eta_4}}\right), \tag{23}$$

  which means the server with a lower value for pheromone deposits is selected for monitoring the selected user. η_3 and η_4 follows the similar properties of η_1 and η_2, respectively. However, for an in-depth evaluation, all the four constants can be varied to check the impact of variation in pheromone deposits and the selection of users and servers.
- ABC-based osmosis: ABC accounts for using three types of bees namely, scout bees, employee bees, and onlooker bees. In the proposed approach, the food sources for which these bees will be looking is the total number of users that can be handled by the available servers. The decision between the scout bees, which are the initial users, and the shortsighted bees, which are the onlooker bees is made by the DSS; whereas the decision between the onlooker bees and employee bees, which are already allocated to servers, is made by the osmotic manager. The users which have a higher fitness value for the food sources gets shifted from being a scout to an onlooker bee, which on the basis of demanded value of a server gets shifted to be an employee bee. The reverse procedures are carried when a user obeys the trust policies in which users acting as employee bees are shifted back to the scout bees. For users to be shifted as onlooker bee, their scout module should have the fitness value B_{so} given as [17]:

$$B_{so} = \max\left(R_{c,t} + M_{c,user} \times \left(R_{c,t} - R_{c,t-1}\right)\right). \tag{24}$$

  and for shifting them from onlooker to employee, the fitness value B_{oe} is given as [17]:

$$B_{oe} = \min\left(R_{c,t} + M_{c,server} \times \left(R_{c,t} - R_{c,t-1}\right)\right). \tag{25}$$

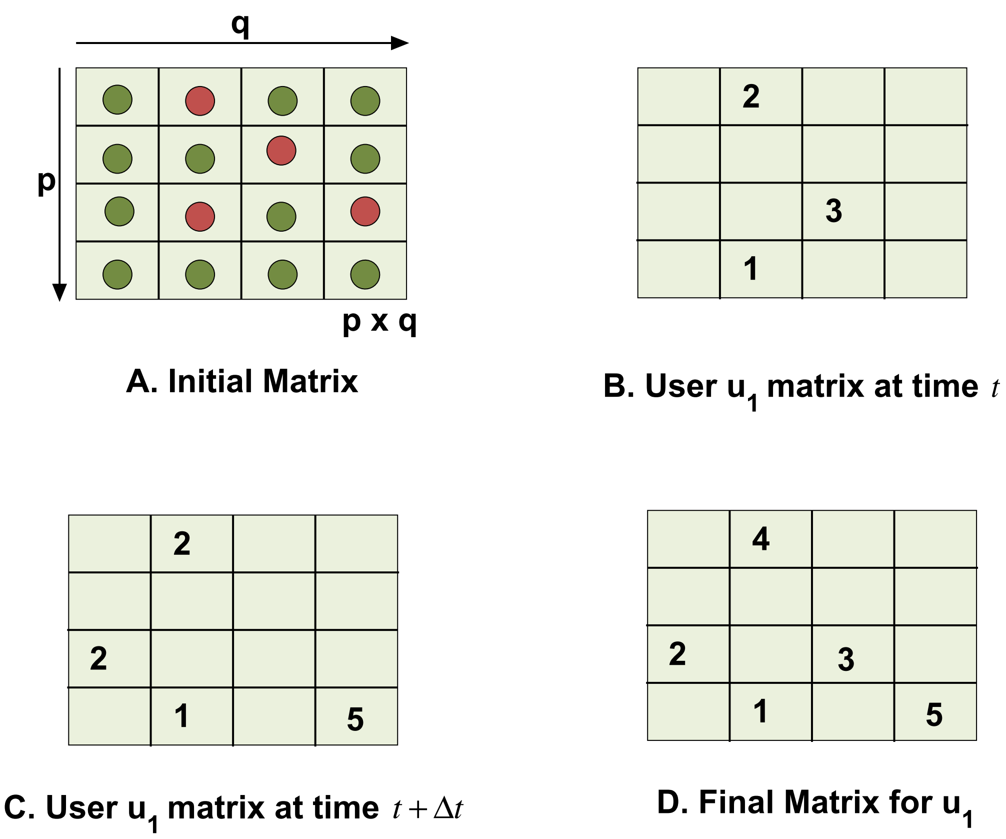
  The reverse over Eqns.(24) and (25) allows users to be shifted back to normal state. The current and previous states considered for evaluation can be replaced by the current and mean values, respectively.

2) PROBABILISTIC OSMOSIS

A fitness-based model requires optimization laws to be obeyed for all iterations, which may cause overheads and may require more number of iterations to arrive at an optimal solution. Contrary to this, a probabilistic model can directly incorporate the FMM to distinguish the users from being monitored or not. The decision on the monitoring of users can be taken on the basis of probability of trust, and the shifting can be done using prediction over probability. The movement of users and selection of servers is done by considering the error in learning over FMM and deviation of an observed value from the predicted value. The users with difference in the mean value greater than the learning value for R_c is moved by the DSS to osmotic manager, i.e., DSS selects users with

$$\begin{gathered}\overline{R_c} \geq P(G_z, G_z|m, z, D_p),\\ \frac{1}{x}\sum_{G_z G_m} \mathbb{P} \geq \frac{\mathbb{P}^c}{\sum_{G_z,G_m} \mathbb{P}^c}, \quad c = 1.\end{gathered} \tag{26}$$

Now, the osmotic manager moves the user with a high difference in the observed value and predicted value to the server with most number of free slots since such user will take more time to balance itself, i.e., osmotic manager moves the user with $\max\left(R_c - R_c^p\right)$ to the server with $\min\left(M_{c,server}\right)$.



🟢	🔴	🟢	🟢
🟢	🟢	🔴	🟢
🟢	🔴	🟢	🔴
🟢	🟢	🟢	🟢

A. Initial Matrix

	2		
		3	
	1		

B. User u_1 matrix at time t

	2		
2			
	1		5

C. User u_1 matrix at time $t+\Delta t$

	4		
2		3	
	1		5

D. Final Matrix for u_1

FIGURE 2. An illustration of lock door policy during post decision making. A) The initial lock door matrix formulated by the osmotic manager which contains a matrix with $p \times q$ mirror links out of which the green one are safe and red one are untrustworthy links. B) A sample of lock door matrix for user u_1 is shown at time t. C) The lock door matrix for user u_1 at time $t + \delta t$. D) Final lock door matrix evaluated at time $t + \Delta t$.

3) THRESHOLD-BASED OSMOSIS

Fitness-based and probabilistic models are highly efficient in terms of accuracy and consideration over the optimization and real-network states. But these models include high computations which may further increase when a large number of users are to be evaluated in POSNs. Thus, a generic and simpler solution can be the matching the observed values against the threshold conditions. However, the selection of an appropriate threshold value is itself a crucial and a highly probabilistic task since no formal approach can allow selection of an appropriate threshold value. Threshold-based movement can be controlled only by varying the parameters, and by defining the upper and lower bounds for each value. In the proposed pervasive model, the trust management and osmosis can be performed together by considering the following conditions:

$$\begin{aligned} D_{u_1,u_2} &\leq D^{TH}_{u_1,u_2}, \\ M_{c,user} &\leq M^{TH}_{c,user}, \\ M_{c,server} &\leq M^{TH}_{c,server}, \\ R_c &\geq R^{TH}_c. \end{aligned} \tag{27}$$

Here, *TH* in the superscript denotes the threshold values. The number of parameters and their threshold can be varied on the basis of current system's state or can be simply considered as the mean value of the particular parameter over the number of states the system has already been through.

C. POST-DECISION MAKING AND LOCK DOOR POLICY

Post decision making is the procedure carried out by the osmotic manager by checking the operational activity of all the users with the help of a lock door policy. The lock door policy is a timely based mirror analyses task which is performed by accounting the number of times a user interacts with a mirrored source. Mirrored sources are those that are not hosted directly over a single server but over a separate web space which may or may not be authenticated by the application hosting server.

An illustration of the lock door procedure during post-decision making is presented in Fig. 2. The osmotic manager maintains a record in the form of a matrix for the number of active links over the application. Every user may or may not interact with these mirror links. The name lock door is derived from the concept of opening a link (door) by the user which it should not open for being safe. Now, depending on the number of visits to these resources, two values, namely, lock-door visits L_v and trust violations T_p are calculated as:

$$L_v(u_1) = \sum_{j=1}^{x} \left(\sum_{i=1}^{cells} \left(\frac{visits}{cells} \right)_i \right)_j, \quad cells = p \times q, \tag{28}$$

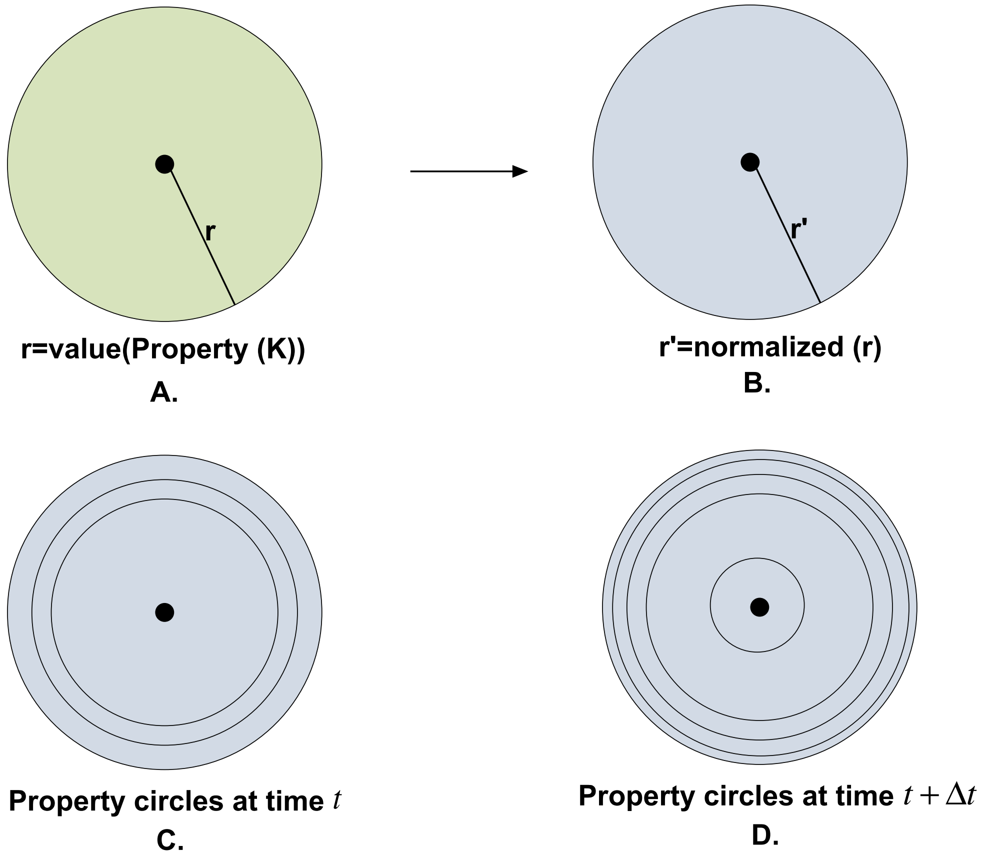

r
r=value(Property (K))
A.
r'
r'=normalized (r)
B.
Property circles at time t
C.
Property circles at time $t+\Delta t$
D.

FIGURE 3. An illustration of property circle formation for a single user. A) An initial property circle with an actual value of the property as a radius. B) An initial property circle with normalized value for a radius. C) The number of circles using normalized values of properties at time t. The number of circles is equal to the number of states after which the properties are viewed. The radius of each state may be less or higher than the previous state depending on the normalized value obtained from the exact values. D) A variation in the number of property circles presented with variation in the number of states at time $t + \Delta t$.

where the number of cells are not static or fixed as the number of mirror links may change with the interval T, *visits* are the total hits on red as well as green mirror links, x are the number of states, and

$$T_{p,t}(u_1) = \frac{visits\_red}{\sum_{i=1}^{N_o} visits\_red}, \quad N_o \leq N. \tag{29}$$

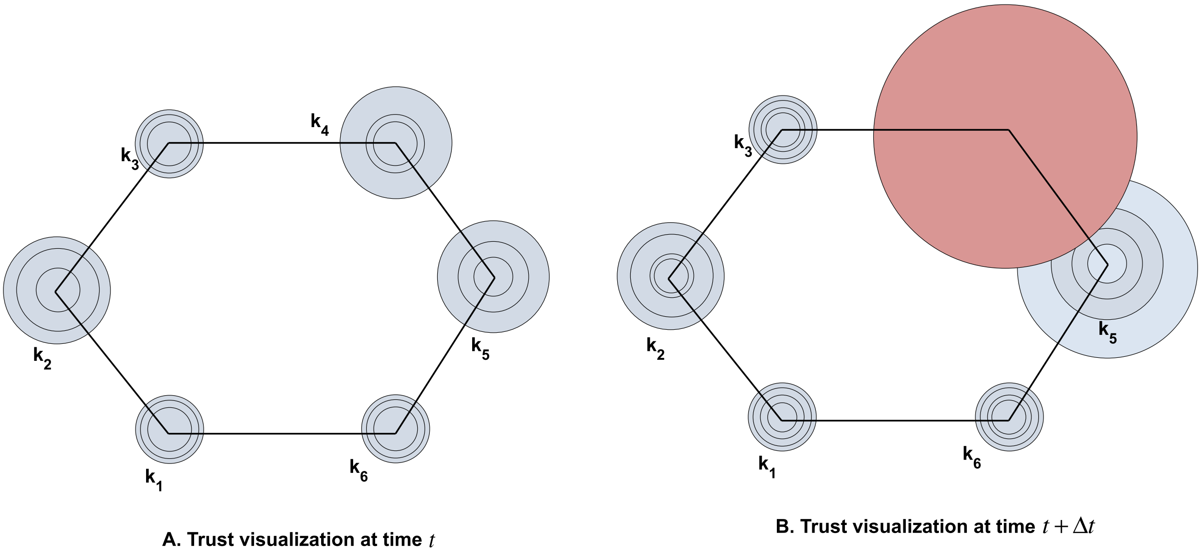
Here, N_o is the number of users under the monitoring of osmotic manager and *visits_red* accounts for visits to untrustworthy links only. The post-decision making over the users' trust can be carried out by using the above parameters only, but using them in coordination with the entire system model allows efficient management of trust as well as faster processing since potential untrustworthy users are evaluated by the separate osmotic cloud. The limits on the lock door values and trust violations can be set by using either of the osmosis procedure defined in the Section IV-B.

D. INTERMEDIATE STATE MANAGEMENT PROCEDURE

Intermediate state management procedure (ISMP) forms the basis of trust-visualization for every user under monitoring by DSS. It allows an intelligent mechanism to understand how the trust of a user varies over time. ISMP allows the formation of a unit radius circle for every property considered in the definition of system model defined in Section III-A. For every property two circles are formulated one with the original value and other with the normalized value as shown in Fig. 3. The inner circles are the values at different states. The number of circles can be reset to control the overlapping of values. The normalization values are calculated using a trivial formula as $\frac{value-\min}{\max - \min}$.

The procedure operates to formulate the visualization mechanism for displaying the trust of every user by using the n-polygon approach, where the number of vertices is equal to the number of properties considered in the initial modeling. The scalability of the polygon formulation allows considering any number of properties which can be classified over POSNs.

In the proposed approach, the visualization is carried using a hexagon as six properties are considered for trust modeling. The procedure includes the formation of a regular hexagon with each side of unit length, and the six properties are placed



A. Trust visualization at time t

B. Trust visualization at time $t + \Delta t$

FIGURE 4. **An illustration of polygon-based trust visualization for a single user. A) Mapping of property circles to the vertices a regular hexagon comprising a set of six properties. The figure illustrates the non-overlapping circles with normalized radii during states at time t. The non-overlapping refers to less dominance amongst the properties. The length of each side of the regular hexagon is 1 as properties are normalized and overlapping can easily be visualized on this scale. B) Mapping results after time $t + \Delta t$ which shows the dominance of property k_4 and k_5. This is an illustration of property for trust visualization irrespective of the order considered during implementation. The number of properties may vary which will vary the type of polygon as well as the order in which these are mapped to the each vertex of polygon.**

on each vertex. The circles formulated over the states are mapped on these vertices which allow checking the variation of properties with variation in iterations as well time. An illustration of the visualization procedure is shown in Fig. 4. The overlapping of the circle accounts for the dominance of a property for a particular user. The visualization process allows monitoring the state of POSNs as well as helps in managing the activity of users by allowing intervention during the intermediate phase of evaluations. The procedure for normalized circle formulation and mapping to a regular polygon allows the formation of an efficient mechanism for intermediate state management during trust evaluation in POSNs. A flowchart representing the implementation procedures is shown in Fig. 5.

V. THEORETICAL AND NUMERICAL ANALYSES

The proposed approach for trust management in POSNs allows efficient control over the entire social network and provides a strategy with low computational overheads, low cost of monitoring and higher computational offloading. This section presents theoretical and numerical evaluations of the proposed model considered for trust evaluations.

- *Remark-1:* Cost of monitoring increases with an increase in the number of unhandled users; and in the ideal case, which includes all the users handled by a single server, M_c is equal to 0; and for limited possession of users, the cost changes to

$$M_{c,server} = N\sum_{i=1}^{N_h}(E_{s,user})_i, \tag{30}$$

Proof: From Eqn. (11), $M_{c,server} = \left(1 - \frac{N_h}{N}\right)\sum_{i=1}^{N_h}(E_{s,user})_i$. Now, for number of handled users equal to the actual number of users, the cost of monitoring will be 0; and for $N_h \ll N$, it becomes $N\sum_{i=1}^{N_h}(E_{s,user})_i$. The increase in the number of unhandled users, the cost of monitoring clearly increases. Note that in normal case of operations, $N_h \neq N$ and $N_h < N$ as a single server cannot handle all the users.

- *Remark-2:* The current value for R_c affects the next predicted value and a higher value causes a decrease in the variation of the predicted value.

Proof: From Eqns. (6) and (7), the increase in R_c gives a lower value for R_c^e which also decreases the mean value causing less deviation in the predicted value as stated in the condition.

- *Remark-3:* There exists a tradeoff between the cost of monitoring and relation cost in POSNs. A highly stabilized and connected social network allows better trust between the users, but this makes cost of monitoring high.

Proof: R_c increases with an increase in the trust value and attains a maximum value when a user satisfies all the conditions given in Eqn. (21). This means that M_c should be minimum for both users and servers as well as the distance of similarity between the probability should be minimum. Now, considering all the fitness, probabilistic, and threshold conditions given in Section IV-B, clearly, there exists a tradeoff between the M_c and R_c, which can be controlled by obeying the conditions defined in Eqns. (14)-(18).

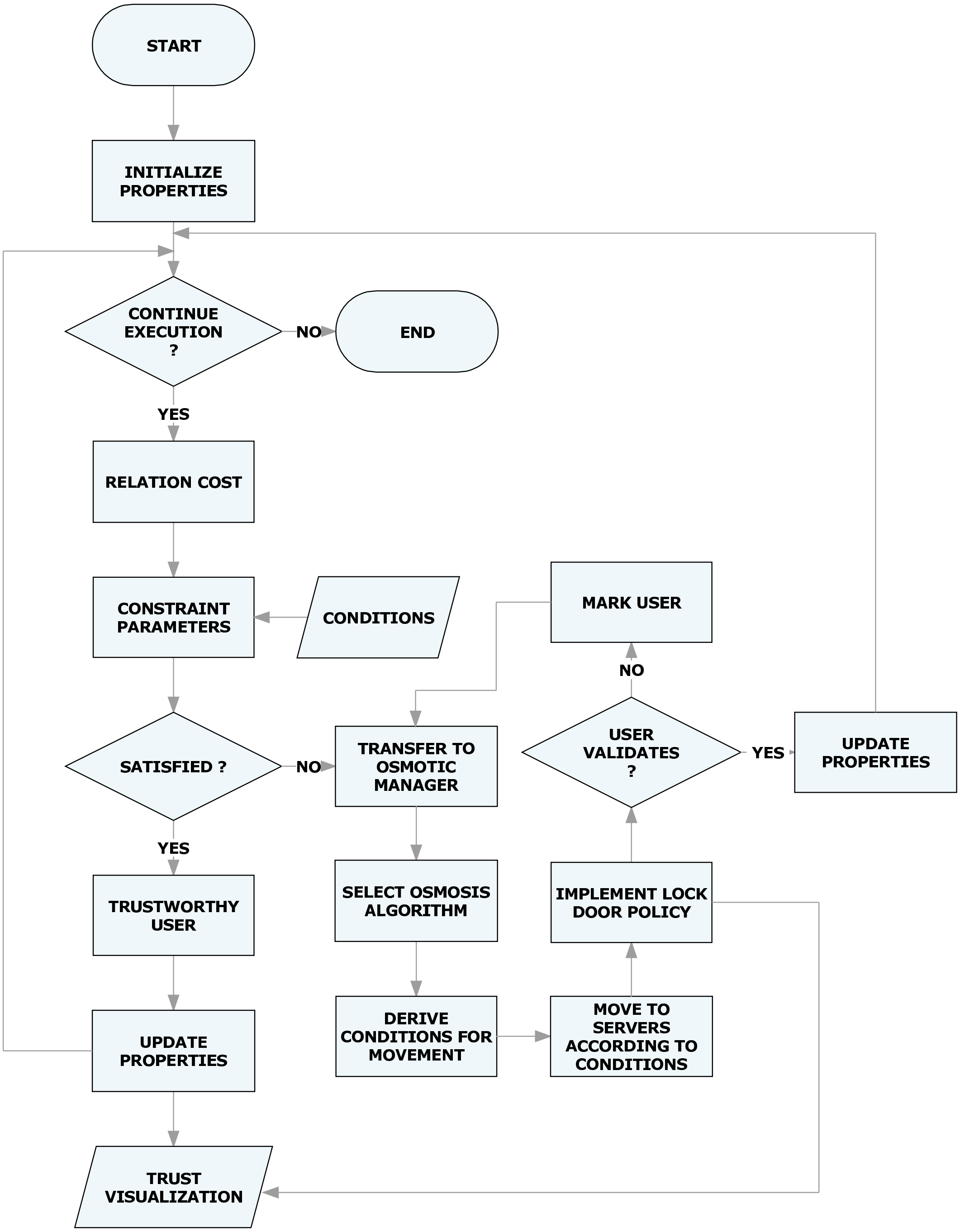


FIGURE 5. Flowchart for implementing the proposed trust management approach.

Next, considering the remarks given above, the proposed approach is evaluated numerically to understand the impact of various metrics over each other. The numerical analyses are carried using pre-derived values of parameters defined in the system model. The number of properties is taken as 6 with the number of states equal to 20, and total users equal to 500. The degree of connectivity is considered by forming a randomized graph with a degree of each user higher than its previous state. A number of servers are considered in such a way that each server can handle at least 50 users. The energy utilization of a user varied between 50kWh and 500kWh with CPU cycles reaching a maximum of 10000 MIPS. The results are presented for average outcomes obtained after 50 successful runs.

Firstly, the results are studied for comparison between the observed relation cost and predicted relation cost of users. Fig. 6 presents the average relation cost for the number of users varying as an independent axis. The figure suggests that the proposed approach is capable of efficiently predicting the next relation cost which can be helpful in finding the

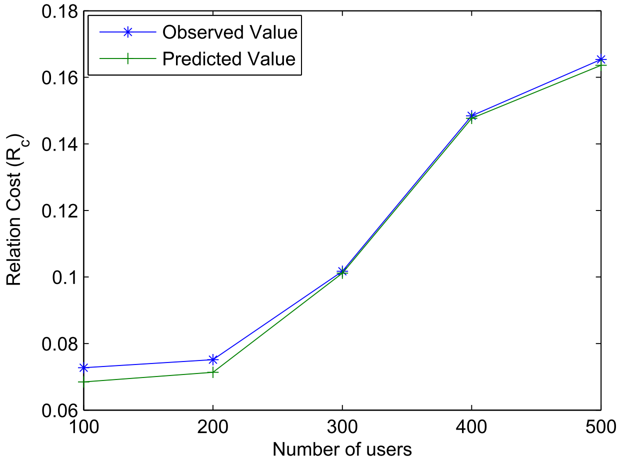


FIGURE 6. Observed relation cost R_c and Predicted relation cost R_c^P vs. number of users.

untrustworthy users across the POSNs. The average error recorded in the prediction of R_c over 500 users is ±2.3%. The lower value can be obtained if a social network is highly connected.

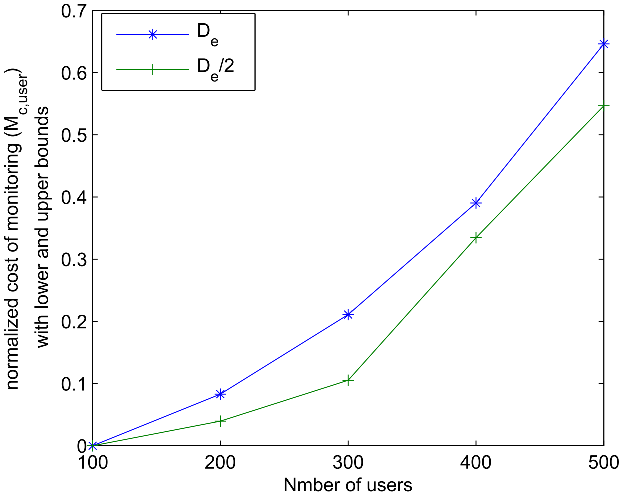


FIGURE 7. Normalized cost of monitoring $M_{c,user}$ with variation in the degree of each user vs. number of users.

Secondly, the results are presented for variation in the average normalized cost of monitoring and average actual cost of monitoring per user by varying the degree of connectivity for 500 users, as shown in Figs. 7 and 8. The cost of monitoring increases with an increase in the number of users since more amount of connections are formed between users which requires more computations to be performed for computing trust scores. However, the degree of connectivity affects the performance of the POSNs, and with a decrease in its value, $M_{c,user}$ decreases.

The normalized value are considered to compare the trend followed by cost of monitoring in comparison to R_c and $E_{s,server}$ as shown in Fig. 9. The graph presents a trade-off between the R_c, $E_{s,server}$ and M_c. It is evident from the graph that more users provide better connectivity in the POSNs and has more depth in connectivity. However, the increase in D_e requires more computations and more analyses over each link, thus, increasing M_c along with an increase in the R_c. This can be controlled by dividing the users across the osmotic layers which allows similar level of trust management with lower energy utilization as well as lower cost of monitoring. The evaluation of the proposed approach utilizing the concept of osmosis is presented in the next section.

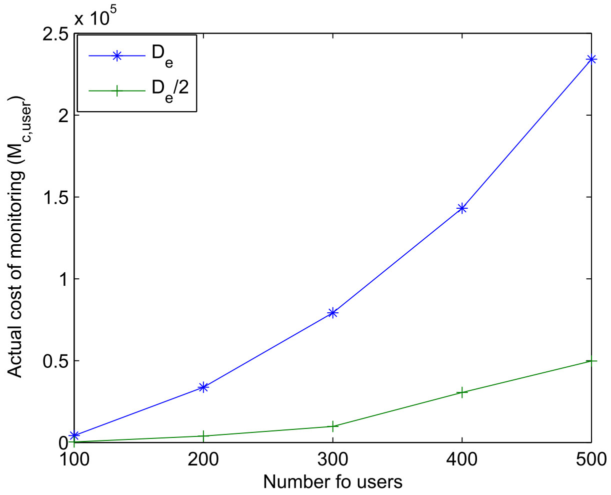


FIGURE 8. Actual cost of monitoring $M_{c,user}$ with variation in the degree of each user vs. number of users.

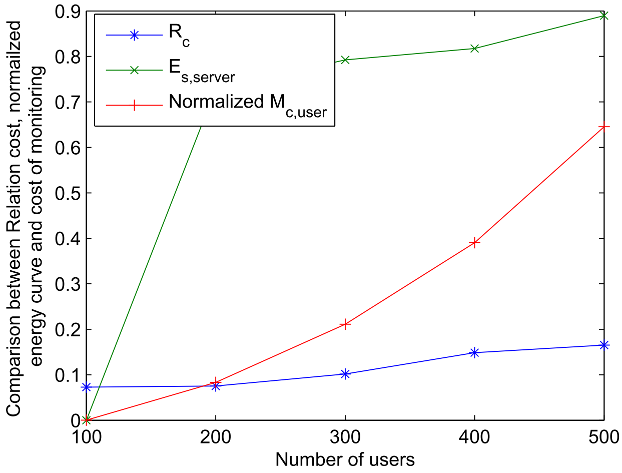


FIGURE 9. Relation cost R_c, normalized cost of monitoring $M_{c,user}$, and normalized energy $E_{s,server}$ vs. number of users.

VI. PERFORMANCE EVALUATIONS

The proposed approach primarily resolves the computations required in managing trust in POSNs by utilizing the features of osmotic computing. The proposed approach is evaluated using simulations conducted in $Matlab^{TM}$. The system model is formulated considering 1000 users which operate over a pervasive environment comprising different layers of servers as explained earlier in Fig. 1.

TABLE 1. Parameter configurations for simulations.

Parameter	Value	Description
N	1000	Number of users
$\|K\|$	6	Number of properties
$\|S\|$	6	Number of sub-classes
V_s	5	Number of servers
V_{os}	5	Number of osmotic servers
V_m	1TB	Memory available per server
$cells$	100	Number of mirror links
red_cells	20	Number of untrustworthy links
C_v	10000 MIPS	CPU cycles
x	50	Number of states observed
$\eta_1 - \eta_4$	1	balancing constants

TABLE 2. Simulation dataset configurations.

Parameter	Value
Users	1000
Maximum Degree	15
Average Degree	8
Maximum Depth	9
Average Depth	6
Maximum Connections	8789
Minimum Connections	8165
Average Connections	8490

The proposed approach is tested on a simulation dataset (provided as supplementary files) generated with the properties defined in Section III-A with configurations given in Tables 1 and 2. The pervasive social network model considered for analyses comprises a user graph with each user connected to other user using a weighted links. The maximum cycles consumed by a user during analyses are 5000 with the minimum equal to 1000. The memory utilization for a user during entire session varies between 5 and 10 GB. The analyses are observed over varying number of states which are represented by the number of iterations. For evaluation of the proposed approach, results are recorded for states that are marked with iterations 10, 20, 30, 40, and 50. The performance of the proposed approach is presented using ACO, ABC, probabilistic and threshold approach in carrying trust evaluations using the concept of osmotic computing.

A. OSMOSIS TIME

It is the measure of time consumed in deciding a server for a user on the basis of its fitness value. The proposed approach primarily focuses on shifting users to the appropriate server for continuous monitoring until the user recovers by improving its relation cost. One of the basic advantages of the proposed approach is the reduction of load over the single server for evaluating trust of every user. The lesser osmosis time justifies the performance of a particular approach for shifting users from osmotic manager to the handling server.

Four different solutions are considered for moving the users which are compared in the Fig. 10. Although threshold and probabilistic approaches are not an optimal solution for the selection of users to be moved between the osmotic servers and osmotic manager, yet these consume lesser time since the movement is merely based on the comparison of two entities. On the contrary, ABC and ACO use fitness score to decide the server and calculate this score each time a user is to be transferred. This adds up to the latency in allocating users to the required osmotic server.

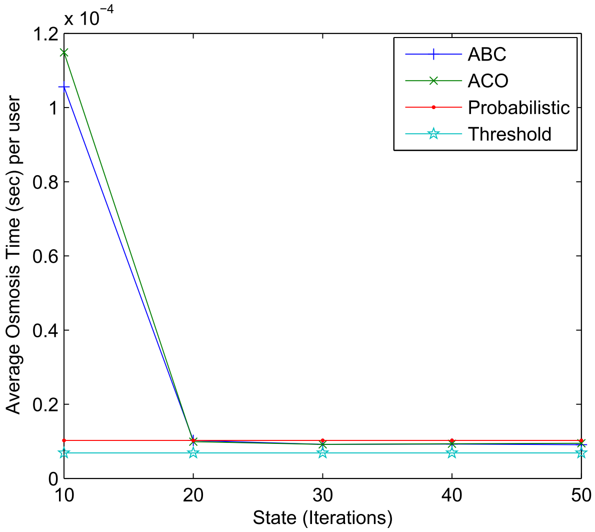


FIGURE 10. Average osmosis time per user vs. variation in state (iterations).

However, after certain iterations, the osmotic time lowers and almost follows a linear trend as the available server obtains a predictive fitness score. The time for osmosis using ABC and ACO decreases lower than the probabilistic approach as this approach is updated each time a user satisfies the constraint over R_c. Further, both ACO and ABC operate differently during initial iterations, but after certain intervals, the time for osmosis overlaps as both algorithms converge to a stabilized state where the users to be transferred are easily predictive by the proposed approach. The results presented in Fig. 10 show that ABC and ACO finally converges and the average osmosis time per user for both is only 17.1% higher than the threshold approach which is quick but not optimal.

B. COMPUTATIONAL OVERHEADS (C_o)

Computational overheads account for the total time consumed by the proposed approach including a decision on the trust of a user and the computations involved in deciding the server for every user. The computational time is much affected by the osmosis time. The proposed approach provides efficient computational offloading which decreases the overall CPU utilization allowing the trust procedures to be completed in lesser time.

Along with the osmosis time, the computations over DSS are also added to obtain the overall computational overheads of the proposed approach. Since the osmosis time is considered in the calculation of C_o, the results are presented for all the four approaches used for osmosis.

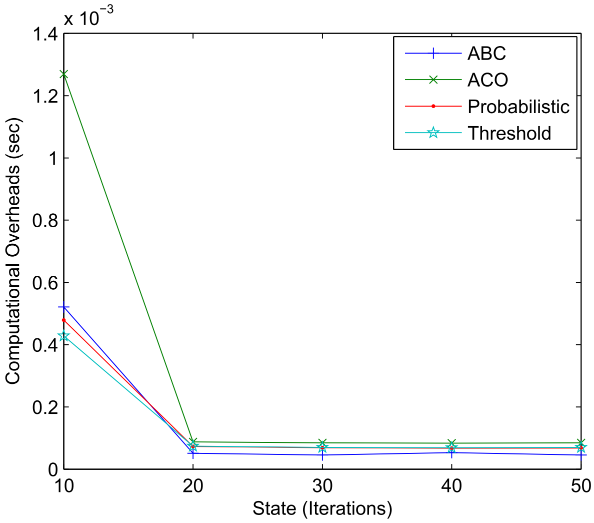


FIGURE 11. **Average C_o per user vs. variation in state (iterations).**

The results presented in Fig. 11 show that the threshold and probabilistic approaches consume lesser time as these do not focus on the optimal solution rather consider simply taking a decision only over the available values. However, unlike the osmosis time, ACO causes more computational overheads than the ABC since ABC divides the users into scout, onlooker and employee bees which reduces the number of computations that are only performed for the scout and onlooker bees; whereas in ACO every user is treated as an ant and the pheromone calculations are performed for every user after each iteration.

Despite the variation in the value of C_o for different algorithmic solutions, the overall overheads of the network are quite low making the proposed approach liable to be used for trust management in POSNs. Although C_o does not guarantee optimality, yet it provides an overview of the latency which might be caused during operations over the DSS and the osmotic manager.

C. RELATION COST (R_c) AND USERS TRANSFERRED

Relation cost forms the base for the activity of the proposed approach to provide computational offloading along with the management of trust. Relation cost is derived on the basis of the probability distribution of various properties considered in the user modeling. A higher value for R_c means lower movement amongst the users across the osmotic servers. A system with a lower value for R_c undergoes more user movement and is not optimal for managing trust despite its lower value for osmosis time.

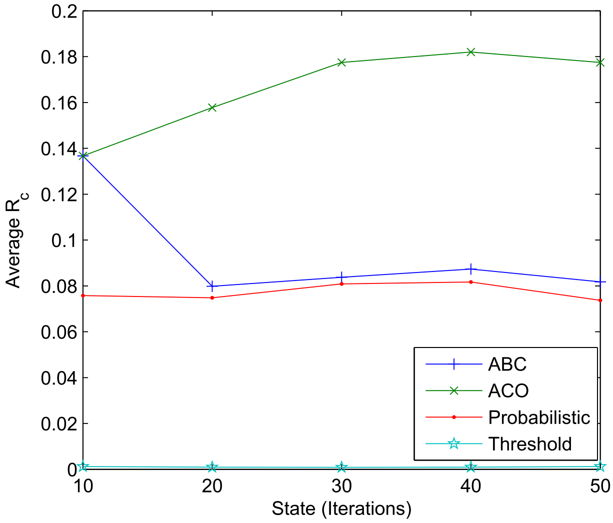


FIGURE 12. **Average relation cost R_c vs. variation in state (iterations).**

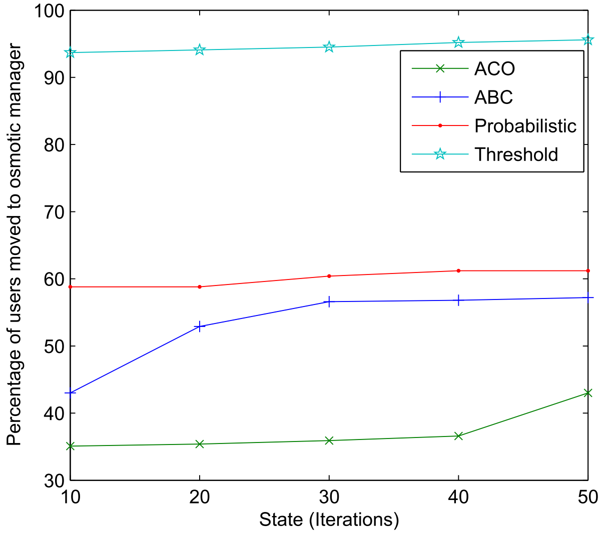


FIGURE 13. **Percentage of users transferred to osmotic servers vs. variation in state (iterations).**

The results for R_c and the corresponding percentage of users transferred across the osmotic manager are presented in Fig. 12 and Fig. 13, respectively. The results show that the threshold approach although consumes less time in deciding the movement of users have lower values for R_c and moves more users to the osmotic servers which increases the computational load over the servers despite a lower value for per user computational time. The increase in load affects the performance of the entire system and causes excessive utilization of the available resources which is against the concept of osmotic computing.

The R_c and percentage users transferred follow an inverse trend. The threshold approach computed lower values for R_c allowing more than 90% of the users to be shifted to osmotic servers. The probabilistic approach also shifted more than 60% of the users to osmotic which adds up to the load over a single server; whereas ACO performed better than all the other three approaches and shifted only reasonable amount of users to the osmotic manager. However, this comes at the cost of excessive computations making ACO an optimal but costly solution for trust management in POSNs.

D. COST OF MONITORING (M_c)

The cost of monitoring is the key for the success of the proposed approach. An approach with a lower cost of monitoring converges faster and provides a solution at a rapid pace. The cost of monitoring takes into account the energy consumed per server, number of CPU cycles utilized, and the variable memory over the servers to which the users are transferred. The normalized comparison is taken to evaluate all the four approaches on a similar scale.

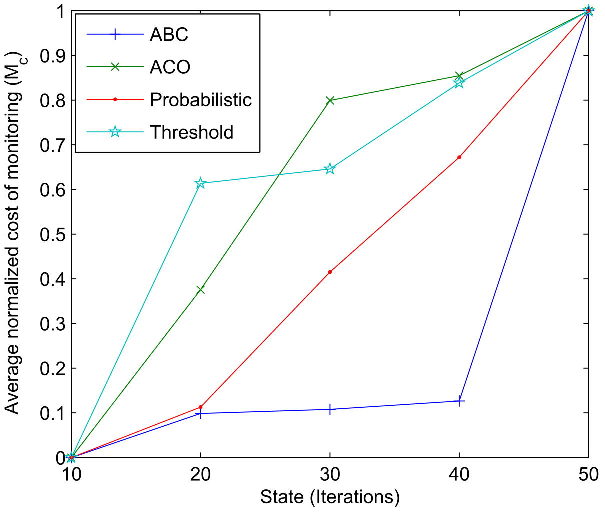


FIGURE 14. Average normalized M_c vs. variation in state (iterations).

Fig. 14 presents the normalized value for M_c over 50 states. ACO performed better for most of the parameters as explained earlier, but it comes with a high cost of monitoring as the calculation of fitness score for every user uses much of the computational and energy resources making it an expensive but efficient solution. Threshold and probabilistic approaches account for an exact fit solution which may or may not work in all scenarios. However, considering the impact of M_c and average performance for other metrics, ABC is well suited for the applications with a bound over its computational and energy resources. ABC can perform better than the other approaches and can maintain a stability by dividing number of users into different bee categories, which reduces the amount of per-user resource consumption.

E. SIMILARITY DISTANCE AND PREDICTIONS

The major constraint in the proposed approach is to maintain a distance of similarity between the relation costs well below the threshold (mean value) set during simulations. The prediction in the R_c comes at the cost of an increase in the average distance of similarity between the users. However, for efficient operations, this distance should always be lower than the threshold value which is set at 1.2 for simulations.

The results in Fig. 15 show that the proposed approach allows efficient prediction of R_c as it is also observed in the numerical evaluations. R_c^p almost overlaps the R_c during simulations generating a minute error of ±1.2%. This error is well under the constraints of that observed during numerical simulations, thus, justifying the efficiency of the proposed approach. The average distance for every user during simulation is lower than the threshold value even after maximum states.

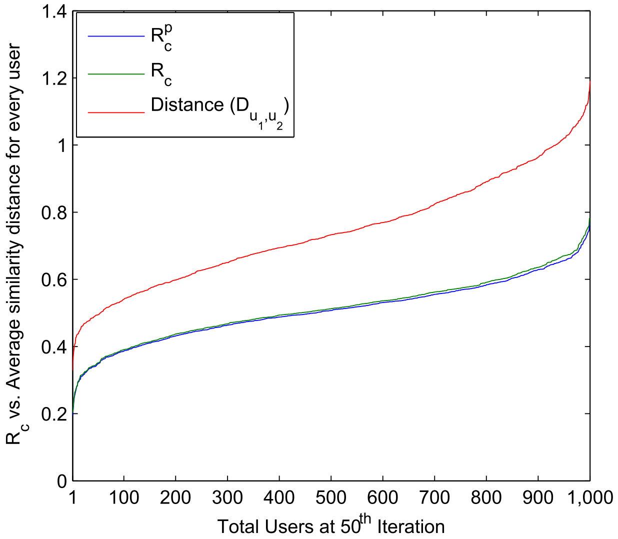


FIGURE 15. Similarity distance, R_c, R_c^p vs. number of users after 50 states.

F. TRUST VISUALIZATION

Trust visualization is a procedure derived using lock door policy and ISMP explained in the Sections IV-C and IV-D. The approach allows analyses of any user during osmosis procedure as well as provides a visualization which allows easy identification of the users that violates the trust policies. An illustration of the trust visualization for the users with ID 500 and 1000 is presented in Fig. 16. The figure presents the dominance of properties considered for calculating the R_c between the iteration number 20 and 50. The blue color represents the controlled properties, green refers to the neutral properties and red refers to the dominating property defined on a normalized scale of 0 and 1. The property visualization over the proposed system is available through dependent normalization of properties of one user over the other. Thus, this system can also identify the most influenced as well as the most influential user in the POSNs.

Clearly, it can be visualized from the Fig. 16 that users with IDs 500 and 1000 have higher value for number of osmotic shifts (L_o), which is a worrying issue for any POSNs. However, after certain iterations and evaluations over the osmotic manager, these users manage to retain their trust policies, and finally reduce their number of osmotic shifts well below 0.5, which is taken as a threshold for all the properties. It can be visualized that relation cost for the user with ID equal to 1000 is more stable than the one with the ID equal to 500 because of a lower number of violations in trust policies. Also, this can be further stabilized by making system undergo more iteration until the user manages to retain the trust policies. The visualization of states allow evaluation of the social network at any state and any user which fails to satisfy the trust properties, and with a

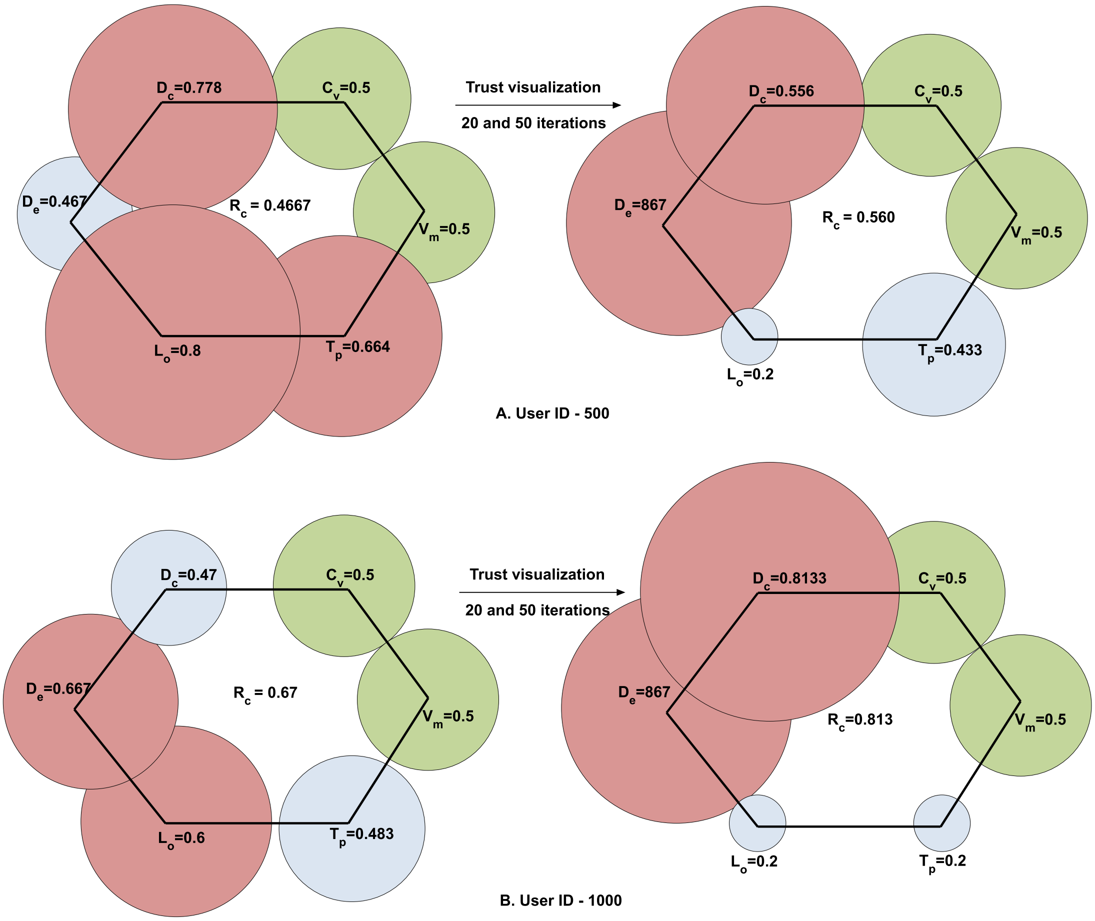


FIGURE 16. Trust visualization with user ID 500 and 1000 between 20th and 50th iterations.

higher value of T_p, can be banned from further operations in the POSNs.

VII. STATE-OF-THE-ART COMPARISON AND DISCUSSIONS

The proposed approach provides efficient trust management in POSNs along with the visualization of trustworthy and untrustworthy users. The proposed approach provides computational offloading by using the concept of osmotic computing. With a lower cost of monitoring and osmosis time, the proposed approach uses a relation cost mechanism to maintain trust amongst nodes of the social network by using four different ways for osmosis. The previous section presented the detailed evaluation of the proposed approach for efficient trust management, visualization, and computational offloading. This section presents a comparative study between state-of-the-art approaches for trust management in PSN as shown in Table 3. From the comparisons, it is clear that existing approaches can provide trust management between the users of a social network, but none of the existing approaches uses computational offloading and cost of monitoring as a key factor in deriving trust values. Thus, the proposed approach proves to be efficient in providing trust management in the scenarios comprising a large number of users operating in POSNs.

POSNs are relatively newer area requiring focus on trust management and behavior analyses of users. There are several key challenges which are yet to be targeted for the formation of efficient POSNs. These include,

- Anonymous authentication while maintaining the trust between the users and abstracted level of visualization: Agreement between the users before starting communications is a key part of POSN. With efficient and swift authentication mechanisms, service dissemination becomes faster, which provides fast and efficient data sharing in POSNs.
- Privacy preservation along with mutual trust management: Privacy should be maintained during all the phases of data sharing. User privacy along with the enhancement of trust improves the operations of POSNs.
- Secure recommender systems for application specific information dissemination: Development of recommender system can identify the state of POSNs and then can formulate an intelligent strategy to counterfeit the

TABLE 3. Existing state-of-the-art approaches for trust management in pervasive social networks.

Approach	Author (Year)	Ideology	Network Type	Evaluation	Computational Offloading	Parameters Improved	Trust Visualization	Cost of Monitoring	Advantages
Secure Pervasive Social Communications	Huang et al. [2]	Attribute-based encryption	PSN	Simulations and Experimental	✗	Operation time	✗	✗	• Less message encryption time. • Less computation cost.
Social behavior study	Zhang et al. [11]	decentralized deep reinforcement learning	PSN	Simulations	✗	decentralized learning, error in estimation	✗	✗	• Better estimation of user patterns. • Flexibility in identifying social behaviors.
Trust management for IoT	Chen et al. [23]	distributed collaborative filtering	SOA-IoT	Simulations	✗	service composition, trust convergence	✓	✗	• High trust convergence. • Highly scalable and accurate.
practical reputation system	Yan et al. [21]	pervasive social chatting	pervasive MANETs	Simulations and experimental	✗	local reputation, robustness	✗	✗	• Useful for end to end applications. • Efficient reputation system.
Pervasive forwarding	Machado et al. [19]	message dissemination	Mobile social networks	Simulations	✗	Average hops, delivery ratio	✗	✗	• High delivery ratio. • lower latency.
PSNController	Ma and Yan [25]	Unwanted content control system	PSN	Experimental	✗	Intrusion detection, transmission speed	✗	✗	• Secure and efficient intrusion detection. • Robust in attacker scenarios.
Driver behaviour detection	Sharma et al. [22]	Driver and vehicle reputation rating	vehicular networks	Experimental	✗	faulty drivers, delivery ratio	✓	✗	• Efficient behavior analyses. • Efficient detection of faulty drivers.
Trust graph generation	Jiang et al. [32]	Trust evaluation using trust graphs	Online social networks	Experimental dataset	✗	trust conflict, trust threshold	✗	✗	• Higher trust accuracy. • Identification of malicious users.
Anonymous Authentication	Yan et al. [10]	Batch signature verification for node trust	PSN	Simulations	✗	Operational time	✗	✗	• Efficient in deriving node trust. • Lower time in signature generation and verification.
Proposed trust management	Sharma et al.	Computational offloading using osmotic computing	POSNs	Simulation dataset	✓	Relation Cost, Average osmosis time	✓	✓	• Efficient trust visualization. • Computational offloading and lower cost of monitoring.

changes occurring due to variation in the properties of users, servers, and connections.

- Efficient device-to-device trust management: The upcoming communication networks include a device to device communication for fast and rapid data sharing. Thus, maintenance of trust between the devices which are operating independently of the infrastructure is a key challenge.

VIII. CONCLUSION

Trust management is tedious in networks operating with a large number of users. Pervasive social networking (PSN) has now evolved as Pervasive Online Social Networks (POSNs) as a large number of users connect through hybrid applications over mobile computing environment. Trust management can be attained by using existing solutions, but these come with a cost of excessive utilization of computational resources making them unfit for real-time evaluations in POSNs.

In this paper, a pervasive trust management framework is presented for POSNs which is capable of generating high trust between the users with a lower cost of monitoring. The proposed approach uses Flexible Mixture Model (FMM) to quantify the system around six different properties, which uses osmotic computing to perform computational offloading for reducing excessive utilization of resources over a single server.

For osmosis, three different solutions are used, which include fitness-based movement, probabilistic movement, and threshold-based movement. The fitness-based movement is generated using the existing Ant Colony Optimization (ACO) and Artificial Bee Colony Optimization (ABC) algorithms.

The novel concepts of lock door policy and intermediate state management procedure are used to allow trust visualization, which provides efficient identification of trustworthy and untrustworthy users. The proposed approach is capable of predicting user ratings efficiently with extremely low errors, which are in the range of ±2%. The evaluations are presented using theoretical analyses, numerical analyses, and simulations. Finally, a state-of-the-art comparison is presented along with discussions and open issues in POSNs.

SUPPLEMENTARY FILES

The datasets and results are provided as separate files.

REFERENCES

[1] Z. Yan and M. Wang, "Protect pervasive social networking based on two-dimensional trust levels," *IEEE Syst. J.*, vol. 11, no. 1, pp. 207–218, Mar. 2017.

[2] C. Huang, Z. Yan, N. Li, and M. Wang, "Secure pervasive social communications based on trust in a distributed way," *IEEE Access*, vol. 4, pp. 9225–9238, Jan. 2017, doi: 10.1109/ACCESS.2017.2647824.

[3] T. Yang *et al.*, "Cooperative networking towards maritime cyber physical systems," *Int. J. Distrib. Sensor Netw.*, vol. 2016, Feb. 2016, Art. no. 3906549, doi: 10.1155/2016/3906549.

[4] S. Zhao and Z. Yan, "Trust evaluation in social networking: A review," in *Proc. 9th EAI Int. Conf. Mobile Multimedia Commun. (ICST)*, Jun. 2016, pp. 58–64.

[5] Z. Yan, R. Kantola, G. Shi, and P. Zhang, "Unwanted content control via trust management in pervasive social networking," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Jul. 2013, pp. 202–209.

[6] G. Gavai, K. Sricharan, D. Gunning, J. Hanley, M. Singhal, and R. Rolleston, "Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, no. 4, pp. 47–63, Dec. 2015.

[7] B. Rashidi and C. Fung, "Disincentivizing malicious users in recdroid using Bayesian game model," *J. Internet Services Inf. Secur.*, vol. 5, no. 2, pp. 33–46, May 2015.

[8] C. Chen, H. Anada, J. Kawamoto, and K. Sakurai, "A hybrid encryption scheme with key-cloning protection: User/terminal double authentication via attributes and fingerprints," *J. Internet Services Inf. Secur.*, vol. 6, no. 2, pp. 23–36, May 2016.

[9] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: Challenges and solutions," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 98–105, Mar. 2015.

[10] Z. Yan, W. Feng, and P. Wang, "Anonymous authentication for trustworthy pervasive social networking," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 3, pp. 88–98, Sep. 2015.

[11] Y. Zhang, B. Song, and P. Zhang, "Social behavior study under pervasive social networking based on decentralized deep reinforcement learning," *J. Netw. Comput. Appl.*, to be published. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2016.11.015

[12] E. Carniani, G. Costantino, F. Marino, F. Martinelli, and P. Mori, "Enhancing video surveillance with usage control and privacy-preserving solutions," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 7, pp. 41–64, Dec. 2016.

[13] Z. Yan, M. Wang, V. Niemi, and R. Kantola, "Secure pervasive social networking based on multi-dimensional trust levels," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 100–108.

[14] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. 20th ICML*, Aug. 2003, pp. 704–711.

[15] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: A new paradigm for edge/cloud integration," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 76–83, Nov./Dec. 2016.

[16] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 11, no. 4, pp. 28–39, Nov. 2006.

[17] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appls. Soft Comput.*, vol. 8, no. 1, pp. 687–697, Jan. 2008.

[18] A. S. McGough *et al.*, "Ben-ware: Identifying anomalous human behaviour in heterogeneous systems using beneficial intelligent software," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, pp. 3–46, Dec. 2015.

[19] K. Machado, A. Boukerche, P. O. V. De Melo, E. Cerqueira, and A. A. F. Loureiro, "Pervasive forwarding mechanism for mobile social networks," *Comput. Netw.*, vol. 111, pp. 6–16, Dec. 2016.

[20] Z. Yan, Y. Chen, and Y. Shen, "Perceptrep: A practical reputation system for pervasive content services," *J. Supercomput.*, vol. 70, no. 3, pp. 1051–1074, Dec. 2014.

[21] Z. Yan, Y. Chen, and Y. Shen, "A practical reputation system for pervasive social chatting," *J. Comput. Syst. Sci.*, vol. 79, no. 5, pp. 556–572, Aug. 2013.

[22] V. Sharma, H.-C. Chen, and R. Kumar, "Driver behaviour detection and vehicle rating using multi-UAV coordinated vehicular networks," *J. Comput. Syst. Sci.*, vol. 86, pp. 3–32, Jun. 2017. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2016.10.003

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

[24] Z. Yan, X. Li, and R. Kantola, "Personal data access based on trust assessment in mobile social networking," in *Proc. 13th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Sep. 2014, pp. 989–994.

[25] S. Ma and Z. Yan, "Psncontroller: An unwanted content control system in pervasive social networking based on trust management," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 1s, p. 17, Oct. 2015.

[26] R. Jin, L. Si, and C. Zhai, "A study of mixture models for collaborative filtering," *Inf. Retr.*, vol. 9, no. 3, pp. 357–382, Jun. 2006.

[27] J. E. Mosimann, "On the compound multinomial distribution, the multivariate ß-distribution, and correlations among proportions," *Biometrika*, vol. 49, nos. 1–2, pp. 65–82, Jun. 1962.

[28] S. V. Nagaraj, "Entropy-based spectrum sensing in cognitive radio," *Signal Process.*, vol. 89, no. 2, pp. 174–180, Feb. 2009.

[29] N. J. Kansal and I. Chana, "Artificial bee colony based energy-aware resource utilization technique for cloud computing," *Concurrency Comput. Pract. Exper.*, vol. 27, no. 5, pp. 1207–1225, Apr. 2015.

[30] M. H. Coen, "A similarity metric for spatial probability distributions," in *Proc. IJCAI*, 2007, pp. 1–6.

[31] A. Baykasoglu, L. Ozbakir, and P. Tapkan, "Artificial bee colony algorithm and its application to generalized assignment problem," in *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, vol. 532. Austria: Itech Education, Dec. 2007, pp. 113–144.

[32] W. Jiang, G. Wang, and J. Wu, "Generating trusted graphs for trust evaluation in online social networks," *Future Generat. Comput. Syst.*, vol. 31, pp. 48–58, Feb. 2014.

VISHAL SHARMA received the B.Tech. and Ph.D. degrees in computer science and engineering from Punjab Technical University and Thapar University in 2012 and 2016, respectively. He worked at Thapar University as a Lecturer in 2016. He is currently a Post-Doctoral Researcher with the Department of Information Security Engineering, MobiSec Lab., Soonchunhyang University, South Korea. He is a member of various professional bodies and a past Chair for the ACM Student Chapter–Thapar University, Patiala. His areas of research and interests include 5G networks, UAVs, estimation theory, and artificial intelligence.

ILSUN YOU (SM'13) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the second Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at the THINmultimedia Inc., Internet Security Co., Ltd. and Hanjo Engineering Co., Ltd. as a Research Engineer. He is currently an Associate Professor at Department of Information Security Engineering, Soonchunhyang University. He has served or is currently serving as a main organizer of international conferences and workshops such as MobiWorld, MIST, SeCIHD, AsiaARES, and so forth. Dr. You is the Editor-in-Chief of the Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications. He is on the Editorial Board for Information Sciences, the Journal of Network and Computer Applications, the International Journal of Ad Hoc and Ubiquitous Computing, Computing and Informatics, the Journal of High Speed Networks, Intelligent Automation and Soft Computing (AutoSoft), and Security and Communication Networks. His main research interests include internet security, authentication, access control, and formal security analysis. He is a fellow of the IET.

Ravinder Kumar received the Ph.D. degree in computer science and engineering from Thapar University in 2015. He is currently an Assistant Professor with the Computer Science and Engineering Department, Thapar University. He is a member of various professional bodies and serves as a reviewer to many referred journals. He has already developed a complete working project on speech recognition and handwritten recognition for Indian regional language (Punjabi). His area of research includes theoretical and practical aspects of combinatorial optimization, approximation algorithm, and mathematical programming.

Pankoo Kim received the B.E. degree from Chosun University in 1988, and the M.S. and Ph.D. degrees in computer engineering from Seoul National University in 1990 and 1994, respectively. He is currently a full Professor at Chosun University. He is an Editor-in-Chief of the IT CoNvergencePRActice Journal. His specific interests include semantic web techniques, semantic information processing and retrieval, multimedia processing, and semantic web and system security.