# Sheffield Hallam University

# A Framework for Exploiting Internet of Things for Context-Aware Trust-Based Personalized Services

OTEBOLAKU, Abayomi <http://orcid.org/0000-0002-4320-9061> and LEE, Gyu Myoung

**Citation:**

*Research Article*

# A Framework for Exploiting Internet of Things for Context-Aware Trust-Based Personalized Services

**Abayomi Otebolaku** [ID] **and Gyu Myoung Lee** [ID]

*Department of Computer Science, Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool, UK*

Correspondence should be addressed to Abayomi Otebolaku; a.m.otebolaku@ljmu.ac.uk

In the last years, we have witnessed the introduction of the Internet of Things (IoT) as an integral part of the Internet with billions of interconnected and addressable everyday objects. On one hand, these objects generate a massive volume of data that can be exploited to gain useful insights into our day-to-day needs. On the other hand, context-aware recommender systems (CARSs) are intelligent systems that assist users to make service consumption choices that satisfy their preferences based on their contextual situations. However, one of the key challenges facing the development and deployment of CARSs is the lack of functionality for providing dynamic and reliable context information required by the recommendation decision process. Thus, data obtained from IoT objects and other sources can be exploited to build CARSs that satisfy users' preferences, improve quality of experience, and boost recommendation accuracy. This article describes various components of a conceptual IoT-based framework for context-aware personalized recommendations. The framework addresses the weakness whereby CARSs rely on static and limited contexts from user's mobile phone by providing additional components for reliable and dynamic context information, using IoT context sources. The core of the framework consists of a context classification and reasoning management and a dynamic user profile model, incorporating trust to improve the accuracy of context-aware personalized recommendations. Experimental evaluations show that incorporating context and trust into personalized recommendation process can improve accuracy.

## 1. Introduction

The Internet of Things is an emerging global Internet-based information infrastructure, which is now positioned as the de facto platform for ubiquitous sensing and personalized service delivery. It promises a new information infrastructure in which all objects around us are connected to the Internet, possessing the capability to communicate with one another with minimal conscious interventions [1–3]. IoT allows people and things to connect at anytime, and anywhere, with anything and anyone, ideally using any network to facilitate global exchange and delivery of intelligent and relevant services [2]. It is the new Internet where things and humans become addressable and readable counterparts [4]. The IoT infrastructure consists of heterogeneous physical and virtual objects that can cooperate on social interactions, where each entity is capable of producing or consuming intelligent services [5]. With this revolutionary and innovative development, it is now

possible for our everyday objects to understand our needs: what we want or prefer and where and when we need them. However, for practical deployment of intelligent applications on this infrastructure, one of the associated problems that need to be addressed is how to find relevant services. A recent and an excellent proposal is context-aware recommendation system that exploits contextual information obtained from devices to learn the user preferences to provide services of interest [6]. CARSs are an extension of traditional recommendation systems, which are generally categorized into three main types: the collaborative filtering (CF) [7], the content-based filtering (CBF) [8, 9], and the hybrid recommendation (HR) systems [10]. On one hand, the traditional CF systems rely on the similarity (or the so-called correlation) between each pair of service consumers, who have consumed or rated the same items, to predict preferences of the target user. If the system predicts the preferences of the target user accurately, then it can suggest relevant and interesting services that the target user has not

yet seen or consumed. On the other hand, the CBF systems use information about the user's consumption history to provide new and relevant services that the user has not yet seen or consumed. The hybrid recommender system is a combination of both CF and CBF, where the strengths of both are harnessed to address their peculiar weaknesses. The most important step therefore in the recommendation process, depending on the recommendation algorithm, is either the determination of the target user's neighbors and the aggregation of the preference information of each neighbor to generate a predicted preference or the determination of services that are similar to the ones the target user has consumed in the past. Note that, in the case of collaborative recommendation, every neighbor must have rated some services in order to participate in the preference prediction process. This means that if none of the neighbors has rated the service, the rating prediction cannot be computed. This problem is referred to as cold start problem [11]. Similarly, one key problem of CBF is overspecialization. An over-specialized CBF system always tends to suggest items or services that are similar to those consumed in the past by the user. However, these problems have been addressed by various excellent proposals [6]. And of late, context-aware recommendation techniques have been proposed and extensively explored with success [6, 12, 13]. The traditional recommendation systems consider user preferences and assume that these preferences do not change as users move from one location to another engaging in various activities. However, in addition to user preferences, CARSs use contextual information such as location, activities, environment situations, traffic information, device characteristics, and network conditions to provide relevant recommendations according to the user preferences in those contextual situations. Despite their widely reported success, CARSs are limited in the type of contextual information they can use and the knowledge they can infer from such context information. In addition, even though CARSs have been explored to understand diverse preferences of users to suggest relevant services [14], nevertheless, apart from the user's mobile phones, some existing CARSs have not taken into consideration other context sources such as everyday objects with which users interact in the IoT environment. Therefore, we argue that existing CARSs lack the adequate capability to provide dynamic and flexible context information required for making dynamic and intelligent recommendation decisions. Thus, the IoT is a novel computing paradigm for developing a new ubiquitous and pervasive network of addressable interconnected heterogeneous everyday objects [14] that can provide dynamic sources of context information. Since IoT objects interact with the environment and other objects, for example, object-to-object and object-to-human interactions sharing vast volume of data, value-added services based on these data can be delivered to users, providing impressive user experience. The IoT possesses the potentials to take context awareness to the next level considering that massive and dynamic data coming from diverse IoT objects can be exploited, via available software interfaces, to build more dynamic and intelligent CARSs [4].

An IoT-based infrastructure is a potential platform to address CARS context problem because it provides an important infrastructure to collect information from various objects that users interact with. This context information can then be exploited to address many challenges of context-aware personalized recommendation systems. There are two main advantages of IoT as a solution to CARS problem. Firstly, IoT provides ubiquitous sensing functionality allowing better understanding of object and human environments. This understanding can be used to gain useful insights into user preferences in a way traditional recommendation approaches, including CARS, have not been able to achieve. Secondly, as illustrated in Figure 1, IoT also allows, for example, collaborative recommendation algorithms, to exploit social characteristics of IoT objects to learn about those objects or humans with similar interests and preferences. However, collecting data, such as context data, from IoT devices poses a new challenge in terms of security and reliability. Thus, trust management is emerging as a powerful tool that can be used to address reliability of not only context data itself but also that of context providers, context information consumers, users, and services [5, 15–18].

The aim of this article is fourfold as illustrated by the high-level functional architecture of the proposed system in Figure 2:

(1) It proposes a conceptual context-aware framework that can collect, analyze, and infer high-level context information from IoT objects. The high-level context is a complex contextual information obtained by combining more than one atomic context using cognitive reasoning or machine intelligence; for example, from "*activity = walking*" and "*location = gym*," a more complex context such as "*walking to the gym*" or "*walking in the gym*" can be inferred.

(2) It provides suggestions to users based on their current and historical context information, taking into account the contextual preferences of the users.

(3) It proposes an extended contextual profile model for entities to manage their service consumption preferences.

(4) It presents how trust can be incorporated into context-aware personalized recommendations.

Thus, in Section 3, the article presents a conceptual framework addressing these fourfold issues. The framework can collect, analyze, and infer context information from mobile sources such as IoT devices. The analyzed context information can be exploited to gain useful insights into the preferences of an entity in an IoT environment in terms of what services they consume or can provide in diverse contextual situations thus satisfying the first aim of the proposed framework. To realize the second aim, the framework provides the components for recommendation processes that can use the inferred contextual information to provide personalized and relevant suggestions. Thirdly, a contextual preference model capable of capturing entity/user preferences, contexts in which such preferences have been expressed, and the level of confidence in such contextual preferences, measured
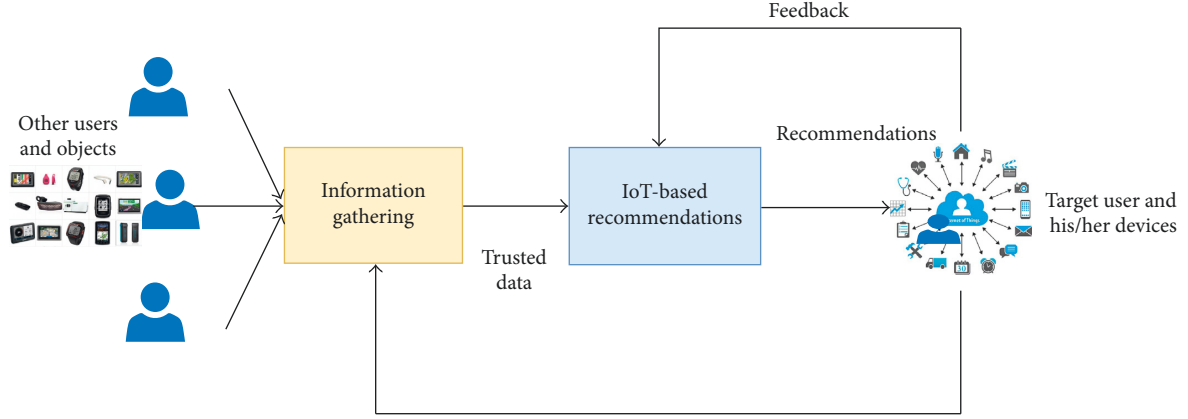
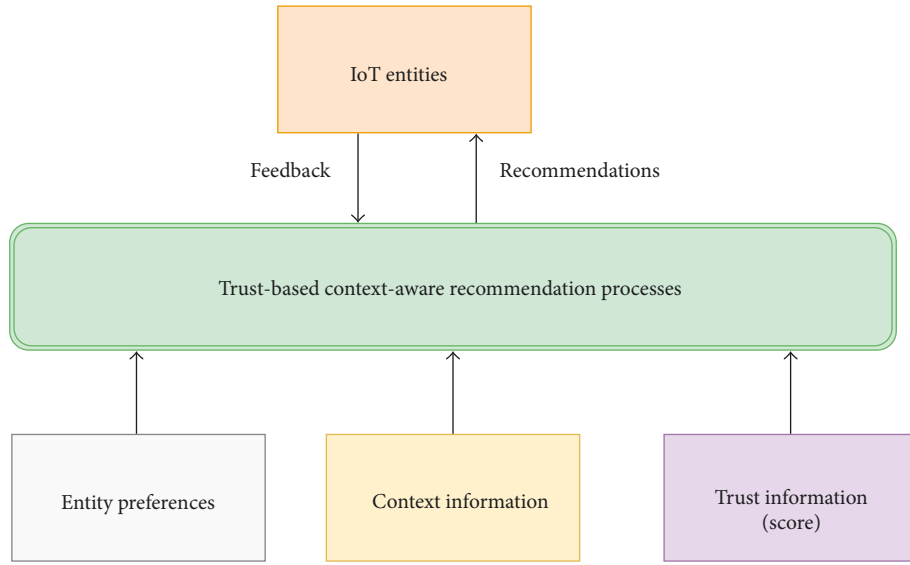FIGURE 1: A high-level view of IoT-based recommendation system.



FIGURE 2: A high-level architecture of the proposed trust-based context-aware recommender systems.

through trust evaluation, is also presented. Finally, we discuss how to incorporate contexts and trust information to improve the relevance of recommendations. The contribution of the current article is a conceptual framework for the provisioning of relevant service recommendations in IoT-based infrastructure.

In Section 2, we discuss background and related work. Section 3 presents the proposed system and its components. In Section 4, we present the experimental setup, evaluations, and results. In Section 5, we discuss the results and provide some insights about our findings. In Section 6, we conclude the article and discuss the future direction of our work.

## 2. Background and Related Work

*2.1. Recommender Systems: Traditional Approaches.* Considering the overwhelming volume of information generated by billions of devices today, the information overload problem cannot be overemphasized [19]. To address this so-called "*information overload problem*," in which information consumers spend invaluable time to find relevant services

due to the humongous volume of myriad available alternatives, personalized recommendation techniques have been extensively explored to deliver such relevant services to users according to their personal interests [6, 13, 20–23]. The traditional recommendation systems take information about items and users and process this information to suggest items of interest to target users. In [24], Resnick and Varian define recommender systems as systems that provide recommendations to people by aggregating and directing them to the appropriate recipients.

Burke [10] define recommendation system as "*any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting and useful objects in a large space of possible options.*" This definition is much broader than the one given by Resnick and Varian [24], in which they essentially define a recommendation system as a collaborative system. In [12], Adomavicius and Tuzhilin provide a formal and widely cited definition of RSs, covering all types of traditional RSs; thus, "*. . . Let C be the set of all users and let S be the set of all possible items that can be recommended. Let u be a utility*

*function that measures the usefulness of item s to user c, that is, u: $C \times S \rightarrow R$, where R is a totally ordered set (e.g., nonnegative integers or real numbers within a certain range). Then, for each user $c \in C$, we want to choose such item $s' \in S$ that maximizes the user's utility.*" Ricci et al. [13] also define RSs "*as software tools and techniques, providing suggestions for items to be of use to a user. The suggestions relate to various decision-making processes such as what items to buy, what music to listen to or what online news to read.*" They have also been exploited to help movie (e.g., Netflix) and music (e.g., Last.fm) lovers make informed decisions on the best and interesting movies or music they prefer to watch or listen to, respectively. Large and leading online industrial service providers such as Yahoo, IMDB, YouTube, and Facebook have deployed RSs as part of their services.

Nevertheless, conventional recommendation systems as defined above have focused on suggesting services to users, based on the information about the service items the user has consumed in the past, designated as content-based filtering (CBF) [9], or using information about those items that other like-minded users have consumed, designated as collaborative-based filtering (CF) [12]. Sometimes, the combination of these techniques, which is designated as hybrid recommendation, is used to push relevant services to users [10]. The core task of a traditional RS is to predict the subjective rating a user would give to an item he/she has not yet seen. Normally, to realize this task, they exploit either the ratings given by friends of the user to services consumed in the past to predict his/her preference for an item he/she has not yet consumed as shown in Figure 3 or they explore the content of the candidate items and the preferences stored in the profile of the target user. Essentially, a recommendation system takes as inputs user information and the description of items and processes the inputs to produce as outputs items likely to be of interest to the user. Additionally, to improve the relevance of future recommendations, a recommendation system learns the user's "*likeness*" (via feedback information) of the recommended items (Figure 1).

Even though recommendation techniques have been developed to provide users with relevant services from very large corpus of information items, however, their main objective is the explicit service suggestion that is relevant to the user preferences; without considering that, such preferences are dynamic and change according to the user's contexts [11, 20, 25]. In addition, the recommendation is made when the user explicitly requests for the RS assistance, and the system does not expect that the user's preferences would change with contexts such as time, activities, and locations. However, users generally prefer certain items in certain contexts. For example, a user who prefers horror and adventure movies would not want to watch horror movies while with his/her two-year-old daughter. Thus, Adomavicius et al. [21] introduced context dimensions into the recommendation systems by extending the traditional 2D (user × item) recommendations to multidimensional recommendation systems (user × item × context), now known as context-aware recommendation systems.

Since the introduction of contexts into recommendation systems, research in context-aware personalized recommendations has explored contexts extensively for its potentials to improve the relevance of recommendations and thus improve the user's quality of experience and recommendation accuracy [6, 11, 21]. Nevertheless, CARSs lack the capability to consider context information from various sensing objects in the mobile environments. IoT devices provide new dimension and capability that can be exploited by CARSs for flexible and dynamic access to various context information characterizing not only the user and his/her consumption environment but also the objects in such an environment.

2.2. *Context-Aware IoT-Based Recommendation Frameworks.* Software frameworks are designed to provide some additional layers of abstraction suitable for specific types of applications [26]. In the IoT environment, there is a limited research work exploiting and integrating IoT and recommendation systems to provide relevant services in this newly emerging ubiquitous environment [4, 14, 23, 27]. Generally, in the design and development of frameworks for CARSs, diverse contextual information has been used, for example, location, time, weather information, environment conditions such as temperature, illumination, and noise level, device characteristics, and network conditions. This information is usually collected from the mobile devices, especially smartphones [20]. However, relying only on context information obtained from mobile phones to provide service recommendations in the ubiquitous environment limits the ability of the system to truly learn the preferences of users as their contextual situations and environment conditions change. Often, the system cannot obtain context information when mobile users do not carry their mobile phones especially at night or when the device has run out of power. Useful contextual information might not be readily captured because in ubiquitous environments, context is dynamic and user preferences change according to the changing contextual situations. Thus, it is important to capture context information from other objects in the user's consumption environments. This gap can be bridged by exploring the potentials of IoT-based context sensing [2]. It allows dynamic, flexible, and seamless sensing of context data whether the user is with his/her device or not. In this section, we review some existing recommendation systems based on IoT context sensing.

In the literature, only limited work is available providing comprehensive frameworks for personalized service recommendations in IoT. The COMPOSE project [28] proposed an innovative framework for building smart and context-aware mobile applications. It utilizes the cloud computing infrastructure and IoT technologies to provide seamless integration of smart objects and external services for delivering scalable resources for data and application management in the IoT environment. Essentially, the framework consists of tools that enable end-to-end development and deployment of context awareness by providing functionality for collecting contextual information on smart devices, functionality for communication with external resources, and an infrastructure for hosting data storage and processing.

In [23], the authors propose a context-aware recommender system framework with a novel temporal interaction
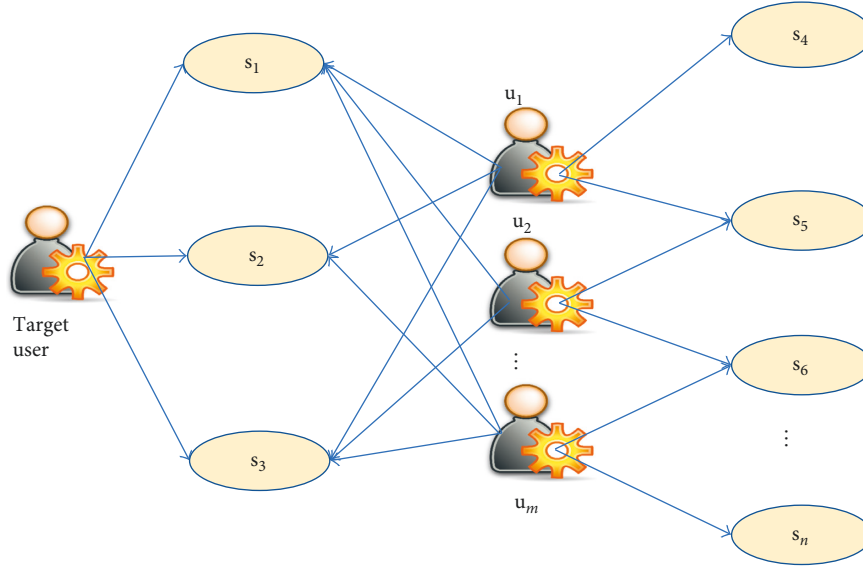
FIGURE 3: Users 1, 2, and 3 are similar to the target user considering their consumption history ($s_1$–$s_3$). Based on this knowledge, CF will recommend a set of items from $s_4$ to $s_n$ (that target user has not yet seen) to the target user.

scheme for IoT-based interactive digital signage, which can be deployed in urban spaces to engage anonymous viewers. The framework was developed using the hybrid recommendation algorithm, combining both content- and collaborative-based recommendation techniques. In the framework, context information is provided by IoT sensing and context data preprocessing modules. What is not clear from this work is how to manage contextual preferences of these "anonymous viewers" to provide personalized recommendations according to their preferences and contexts.

Among other proposals that attempt to address the concern in [23] is the one presented in [14] by Mashal et al. They proposed a recommender system for suggesting services, which are relevant to users' preferences according to the contextual information obtained from IoT objects. They developed a weighted undirected tripartite graph-based service recommendation algorithm for delivering contextual personalized services. Another interesting and an excellent work with similar direction is the one presented in [4] by Hussein et al. They have proposed a social Internet of Things (SIoT) based intelligent recommendation framework that was developed on top of the SIoT context infrastructure, which uses cognitive reasoning mechanisms on context elements obtained from SIoT sensing objects. They developed ontologies for inferring user or object contextual situations. They emphasized and developed a task navigator algorithm that matches situational goals against the quotidian task ontology to determine which tasks match the situational goals and which smart services can fulfill such tasks. Similar to this proposal, the main goal of our proposal in this article is to provide a generic framework for personalized service recommendations, adapted to the target users according to their contextual situations and environment conditions obtained from IoT devices.

In [27], Saleem et al. proposed a concept for exploiting the SIoT for service recommendations among various IoT

applications. The authors highlight how the SIoT can be used to provide recommendations by presenting an interesting sample application scenario. Whilst being an excellent proposal that intends to exploit opportunities provided by the thing-to-thing and thing-to-human social relationships of the Internet of things, it is not clear how they will exploit IoT and CARS to provide context-aware personalized recommendations. In [22], the authors proposed a novel algorithm for providing venue recommendations that include information sourced from IoT, Web services, and applications, as well as social networks (SNs) in the context of IoT. The proposed algorithm, according to the authors, also considers qualitative attributes and semantic information of the venues (e.g., price and atmosphere), the profile and habits of the user for whom the recommendation is generated, and the opinions of the user's influencers.

These solutions are some of the excellent proposals using IoT-based contextual information to develop intelligent and adaptive service recommendations. These proposals show that recommendation systems are effective tools for solving information overload problems, especially in the newly emerging IoT infrastructure. However, judging from the analysis above, some key issues still need to be addressed. Apart from some areas pointed out above, these systems lack the capability to improve future recommendations to users based on their experience of the current consumption. It means they lack the means to collect feedbacks from users and use these feedbacks to better understand the user preferences. Thus, the conceptual framework developed in this article provides a bridge between those proposals that do not provide means to personalize and improve service recommendations by developing a reliable contextual profile model able to learn user preferences from their ever-changing contextual situations, environment conditions, and feedbacks. The framework has been designed not only with the flexibility to allow implementation of new recommendation algorithms but also

to accommodate the extension of the traditional recommendation algorithms by using IoT-based context sensing. It also proposes the incorporation of a trust management component to enforce the reliability of context information sources as well as those users who consume services provided by the system. Thus, instead of having the CARSs with (*user* × *items* × *contexts*) *dimensions*, we now have *user* × *items* × *items* × *trust* dimensions.

## 3. Proposed System Framework

*3.1. Overview of the Proposed System Architecture.* In this section, we present an overview of the proposed conceptual framework for context-aware personalized service delivery in IoT. The predicted deployment of billions of devices and their global access require layered architecture for the management of information from the physical devices to the cloud computing [1–3]. IoT will take the concept of Ambient Intelligence to the next level as it supports the concrete realization of building a digital ecosystem that is fully aware of the environments without conscious human intervention [14, 26, 29]. To realize such technological development, some key features must be present: (1) the IoT-based systems should be adaptive, and they should change in response to contextual situations. (2) They should be dynamic by anticipating preferences of entities or users. (3) They should be context aware to recognize the environment, objects, or entity's situational contexts. (4) The IoT-based systems should support tailoring of services to the users' contextual preferences, and (5) the sources of information and information itself must be reliable. Thus, we propose trust as a means for improving the reliability of personalized recommendations by using feedback from users to compute trust after consuming services in specific contexts.

Firstly, we give a high-level overview of a typical IoT platform, such as the one being developed by the Wise-IoT project consortium [30]. Wise-IoT project is an EU H2020 funded project, being executed by leading European research and academic institutes and their counterparts in South Korea. Wise-IoT aims to provide a worldwide interoperable IoT that utilizes a large variety of different IoT systems and combines them with contextualized information from various data sources. Such an IoT platform can be viewed broadly as a three-layered architecture as illustrated in Figure 4. Thus, the first layer of the architecture is the sensor or physical layer where many heterogeneous IoT devices are deployed. These devices can communicate with each other, using standardized connectivity such as Wi-Fi, 3GPP, 3G/4G, Bluetooth, ZigBee, and LoRaWAN (supporting low-power connectivity) based on such standards as OneM2M. The OneM2M standard is a global IoT standard that addresses communication issues for a common IoT service layer and interoperability on the IoT connectivity layer [31]. This sensor layer is also responsible for the IoT management.

The second layer is the contextual information management layer and can be considered as the cloud layer responsible for the acquisition, aggregation, and processing of context data. Furthermore, low-level context data from the physical layer are preprocessed for cognitive inference
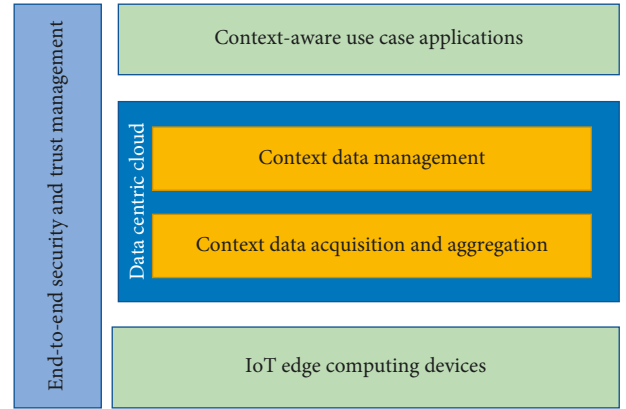


FIGURE 4: IoT context abstraction layers.

purposes using various techniques such as data mining or machine-learning algorithms as well as semantic web models for context reasoning, consisting of a context knowledge base and query and inference engines [32, 33]. One of the current and popular implementations of this layer is the FIWARE context broker, which is based on the NGSI standard [17]. FIWARE provides the capabilities to obtain IoT context data in addition to context information obtained directly from the user's mobile devices. Nevertheless, presently, these context brokers such as the one developed in the FIWARE project, do not provide the capability for context classification and knowledge processing and logical inference [31]. The last and the upper most layer is the application layer where context information is exploited to deliver intelligent services. Different kinds of applications can be deployed at this layer, for example, car parking or route recommendation applications.

Similarly, the framework proposed in this article is a three-layer architecture as illustrated in Figure 5. The three layers align with the two layers of Figure 4. The first layer in Figure 5 consists of those components that provide the functionality for dynamic context sensing, context recognition, context information distribution, cognitive context reasoning, and context information model. In the second layer, we have the context-aware trust-based personalization-related components. Among these components are the trust (evaluation) manager, which is responsible for computing trustworthiness of an entity of the platform [16]. We also have a contextual profile model, recommendation manager, feedback manager, and contextual profile manager. The contextual profile model is responsible for learning the entity's preferences in contexts of service consumption. The recommendation manager is responsible for using the contextual preferences to filter candidate services and suggest the most relevant ones to the users. The recommendation manager manages the selection of recommendation techniques or algorithms to use during the filtering process. The feedback manager is responsible for using implicit or explicit feedback obtained from users after consuming recommended services. The IoT context information access and processing (CIAP) layer comprises various components for context sensing, recognition, cognitive context reasoning, and context broker for context distribution.
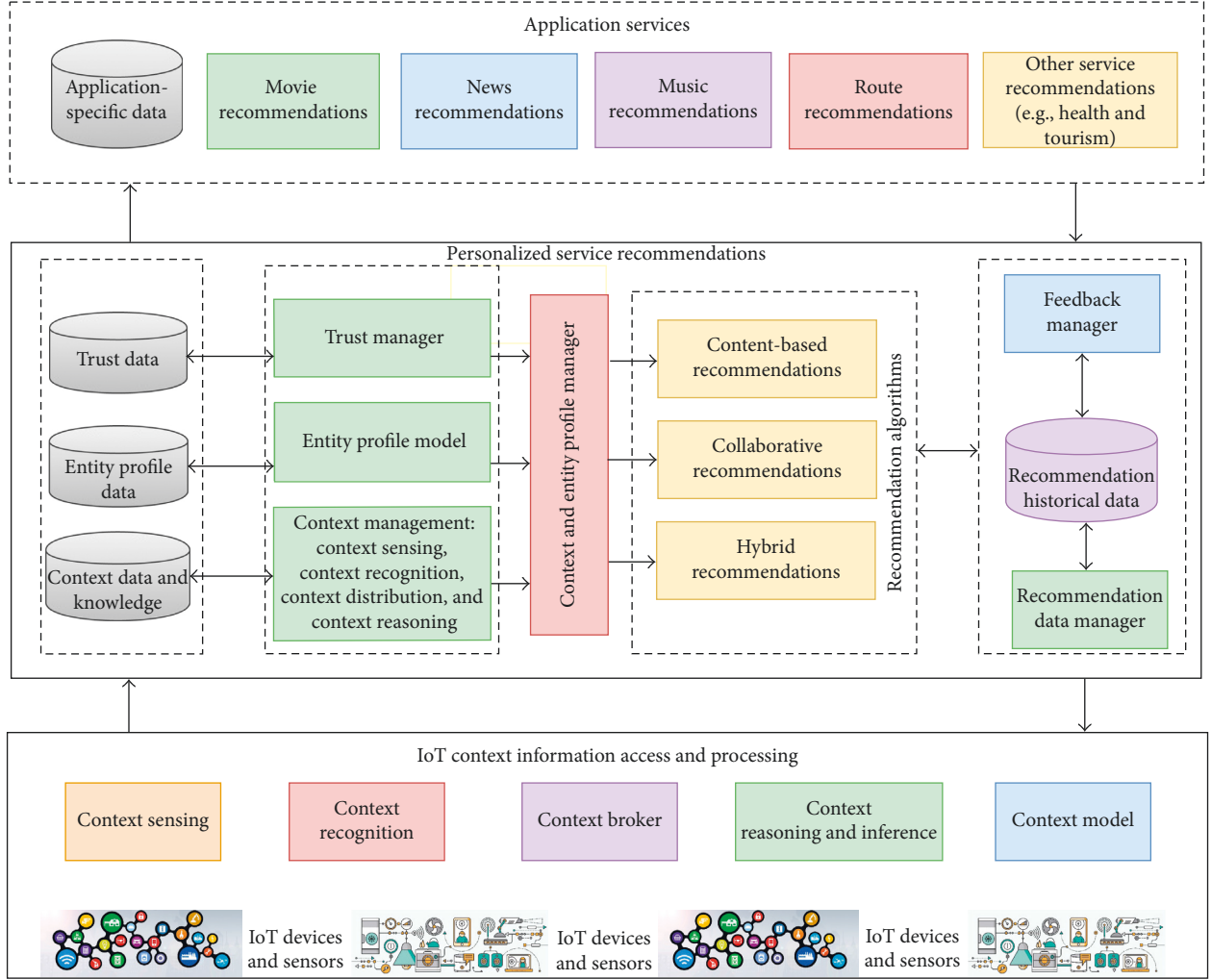
FIGURE 5: Components of the proposed trust-based context-aware recommendation system.

*3.2. Context Information Access and Processing.* This section introduces the context awareness component (as illustrated in Figure 6) that we have designed, for deployment on the IoT platform to process context data obtained from IoT context brokers and other IoT platforms such as the FIWARE broker [31]. Each of these components will be discussed in the next sections. The framework consists of various processing modules to be incorporated into our trust-based platform for intelligent service delivery in the IoT environment [16]. The context framework has been designed to provide the capability for high-level context classification, using machine learning and ontologies for context classification, semantic processing, and reasoning. In [34], we have described in detail context sensing, context classification, and reasoning using the neural network algorithm and various statistical features to identify the context in IoT. In this section, we only provide various steps for context classification and reasoning. However, we extend the concepts developed in that paper by providing here the cognitive model of context information and inference using ontologies, reasoners, and Semantic Web Rule Language (SWRL) [35]. Figure 7 illustrates the relationships between IoT

sensing, cognitive context management, and intelligent IoT-based applications. Figure 8 illustrates an example of how the context framework can infer high-level contexts (watching TV) from raw data accelerometer, GPS, and so on directly obtained from sensors.

*3.2.1. Context Sensing and Preprocessing.* The ability of a system to identify contextual situations and to make informed decisions is one of the most important functions of any context-aware system. However, sensors emit data that are in low-level formats, which are not suitable for decision-making by mobile applications [36]. Context recognition process collects raw data from sensors and transforms them into information that can be used to build intelligent applications. To provide an accurate context information about service consumers, the proposed framework uses the context recognition process to identify contextual information such as user activities, from smartphone-embedded and IoT sensors (obtained in the form of Web Services APIs from IoT platforms). The framework's sensing module gathers events from these sensors such as accelerometer, gyroscope,
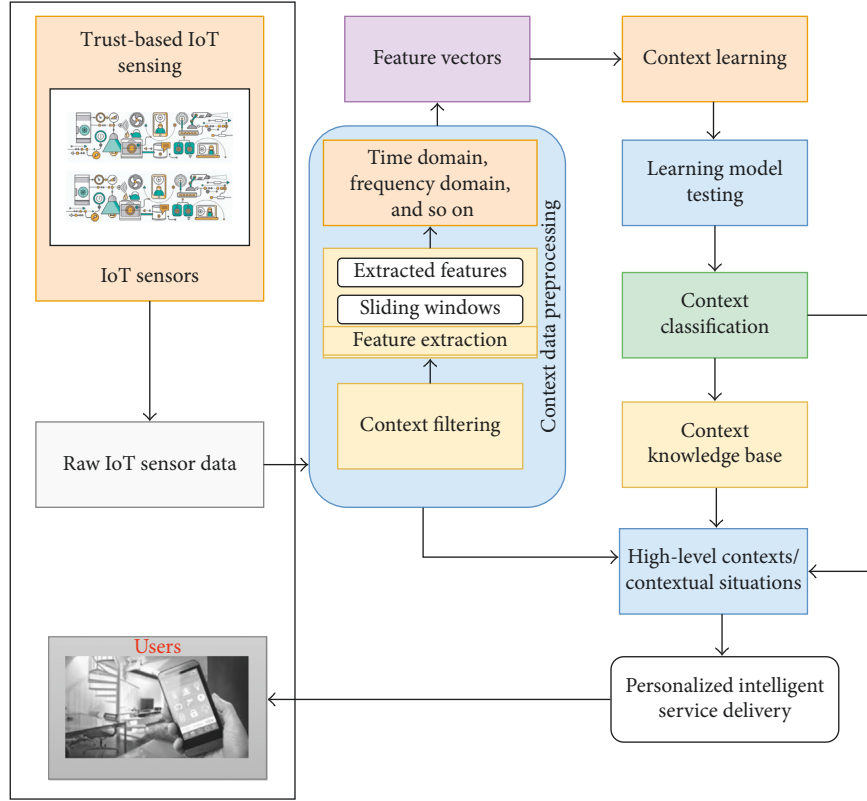
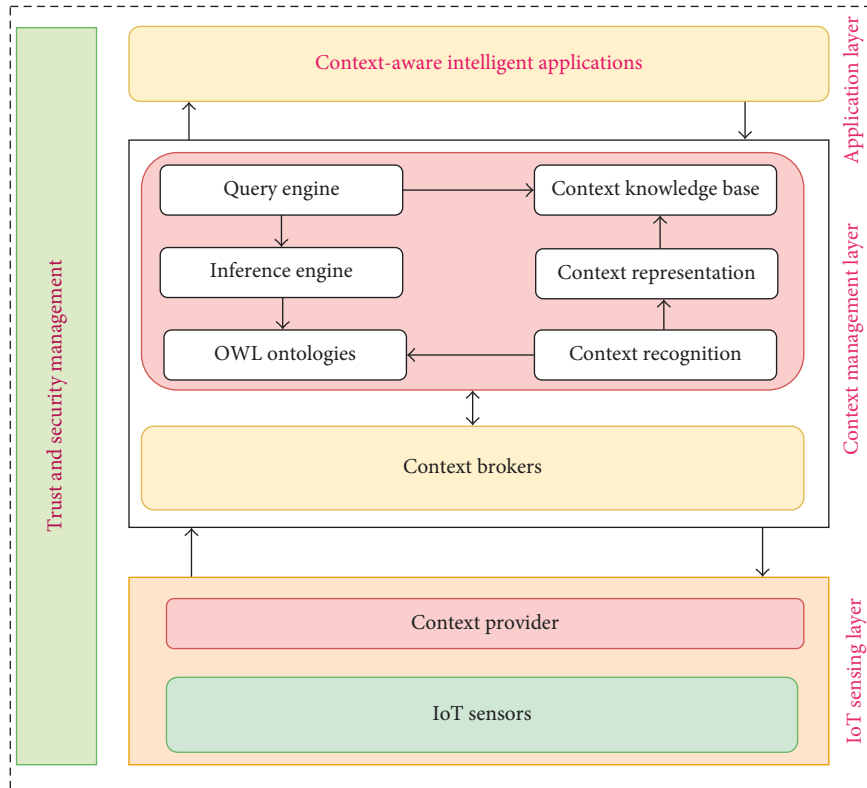Figure 6: Context recognition processes of the proposed framework.



Figure 7: High-level architecture of the proposed cognitive context awareness, an extension of Figure 6.
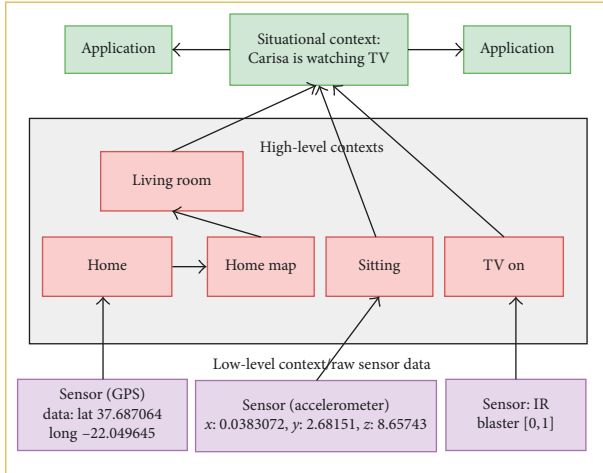
Figure 8: Situational context from high-level context and low-level context data.

rotation vector, orientation, proximity, microphone, light sensors, and other sensing devices, preprocesses the data, and then uses classification algorithm(s) to derive a more meaningful context information. It collects sample data from each sensor in predetermined time intervals overlapping the next predetermined time space. In the data-preprocessing phase, the framework removes the data outliers [34, 36]. This is achieved by eliminating samples from the beginning and ending of each example to reduce the influence of noise in the streams of sensor data.

*3.2.2. Feature Extraction.* The user's dynamic contexts, such as activities, occur in relatively long period, in seconds or minutes, compared to the sampling rate of the sensors. This sampling rate usually does not provide sufficient data to describe user activities. Therefore, activities are usually identified on time window basis rather than sampling rate basis [37]. Comparing a series of time windows to identify activities is almost impossible even if the signals being compared come from the same user performing the same activity context [13]. The feature extraction process addresses this problem by filtering relevant data and obtaining quantitative data from each time window. Many approaches have been explored in literature such as statistical and structural properties of the sensor signals. Structural features such as Fourier transform are quite complex and require more computational resources [37]. This may not be ideal for an energy-starved smart device. On the other hand, statistical features are simple and require less computational resources. Thus, this process uses simple labeled statistical features (e.g., range, maximum, minimum, mean, and standard deviation), which have been validated in our previous work to be very effective to discriminate between time windows [34, 36]. These features are extracted into feature vectors that are then used in the next process.

*3.2.3. Context Classification.* After extracting the time window features from the raw sensor signals, without deriving the context knowledge from them, the example features are limited in how they can be used. The context

classification uses supervised machine learning algorithms such as the neural network (NN) to derive high-level context from the statistical features. Details of the modeling and evaluations can be found in [34]. These models are integrated into the recognition service to obtain independent future activities and contexts of the users.

*3.2.4. High-Level Context/Contextual Situations: Context Inference.* The context information obtained from context recognition and classification processes is not able to provide cognitive meaning of complex context information, in addition to not being able to recognize complex context information. Thus, it is important to develop a cognitive model based on semantic web technologies that can infer higher-level contextual situation from these individual contexts as illustrated in Figure 7. For example, it is important to know what a user is presently doing, when, and where he/she is doing it. Nevertheless, without relating this information to provide a more high-level contexts, this information will not be useful to the application. This is one of the weaknesses of some of the existing work. The proposed context framework thus uses the knowledge-based model on top of the classification process to relate different atomic context information to obtain contextual information at a higher semantic level. For example, having known that a user sitting at home is in the living room, if we know that the TV is on, then it is important to relate this information and conclude that the user is watching TV. For example, having known that a user located at home is sitting in the living room, if we know that her TV is on, then it is important to relate this information and conclude that the user is watching TV. The user is watching TV, obtained from sitting, home, living room, TV, and so on, is inferred using the knowledge-based cognitive process. The details of this process can be found in [35, 36]. It can also determine complex context such as a user is "*jogging at the sport arena.*" This kind of information is crucial to offering rich quality of experience to users.

*(1) Ontology Concepts.* In this section, we introduce the design of the ontology model in the proposed framework for the inference of higher or complex contextual information from the atomic context classified by the context recognition model. To infer higher-level or complex contextual situation from the recognized and classified contexts from the previous section, we have developed semantic web-based ontology reasoning for context in the IoT environment. As illustrated in Figure 9, we have provided some important concepts from which the ontology was designed and developed. To infer high-level contexts from those concepts, context reasoners and context rules have been used. The conceptual context framework incorporates nine generic interrelated concepts, which were defined for providing context information for the context-aware media recommendation domain. Eight of the core concepts are based on our previous context representation model in [35] but now extended to incorporate trust as an additional concept that relates with other core concepts as a property.

The ontology is being developed using OWL DL with adequate provision for adaptation decisions, relying on
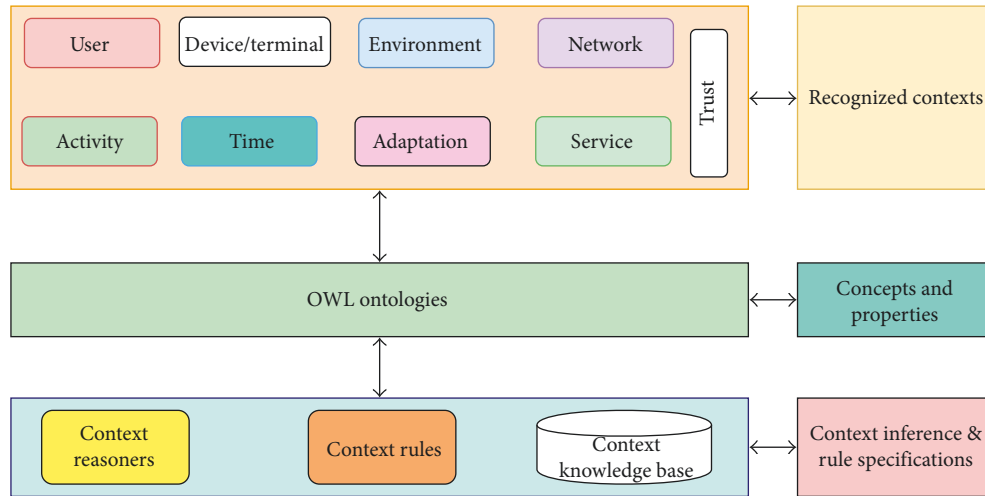
FIGURE 9: The framework inference model.

various rule-based mechanisms, which support an automatic knowledge inference from contextual information [38, 39]. The cognitive model has been designed as a two-layered model consisting of lower and upper layers representing upper and lower domain ontologies as illustrated in Figure 10. The first layer is the primary or upper domain ontology with 9 core concepts: *User, Device, Environment, Service, Activity, Time, Trust, Adaptation, and Network*. The concepts in this domain are generic and are valid for many application scenarios. The second layer incorporates specific domain ontologies, integrating reasoning and rules for inferring contexts specific to each domain. The specific or lower domain ontology is designed based on the combination of user specific activities and locations and other contextual situations. For example, the user *home domain activity* and the related contexts determine contextual information that characterizes the user's activities when at home. Figure 11 illustrates the relationships between semantic concepts, the building blocks of the proposed cognitive context reasoning model, ontology, and the SWRL-based context model. We define the primary concepts of the context model as follows:

*User*: the central focus of any context-aware recommendation system is the user. In these systems, information such as location, activities, and preferences is very crucial to building a system that adapts to the user's needs. This contextual information influences the user's preferences for service consumption, and therefore, when recommending service items to the user, it is important that this information be incorporated into the recommendation process.

*Time*: the time concept is primarily used to capture the "when question" of the other concepts. Time duration, start time, end time, and so on are some of the elements of this concept. These elements of the time concept are used to infer, for instance, high-level time information such as weekdays or weekends which is subsequently used in addition to location information to infer a user's contextual situation including her activity contexts.

*Device*: the characteristics and capabilities of IoT devices can have significant impact on the personalized

recommendation process. The device characteristics can be used to adapt recommendations to the user's device.

For example, the device battery level can be used to determine the appropriate format to present recommended items. In fact, it may present the item such as streaming video in a format that requires less power such as in text form. If the battery level is low, the recommendation system could decrease the brightness or lower the spatiotemporal resolution of the content in case of an audiovisual content. In addition, this concept also describes the capabilities of the devices.

*Environment*: another important concept of the ontology is the natural environment. This concept influences the kind of items users would prefer to consume depending on the environmental condition or situational contexts. It provides information about the environment in which a user interacts with the recommendation system or his device. There are two groups of fundamental concepts related to the environment concepts: location, including logical (e.g., city and street) and physical (e.g., GPS coordinates), and environment conditions (e.g., noise level, illumination, and weather information). For example, information about the current weather can be used to provide more relevant media recommendations to users, that is, the category and genre of contents preferred by the users can be influenced by weather information.

*Activity*: user activities can be used in addition to the other context information by the recommendation process, given that they have a strong impact on the preferences and needs of mobile users, as previously stated. As explained in the previous sections, to address this problem, low-level data obtained from device-embedded sensors are channeled to a context recognition model running on the device, which recognizes high-level activity being performed by the user. To relate this activity information, as illustrated in Figure 12, with other contextual information, an activity concept was designed as part of the ontology.

*Network*: the network concept describes the conditions and characteristics of the network connection of the user
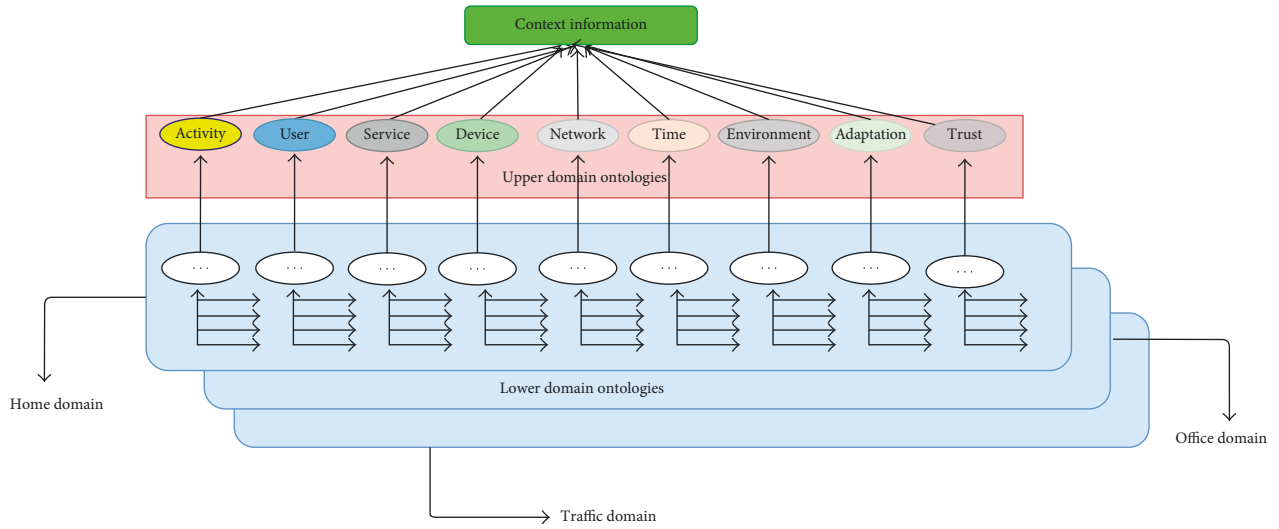
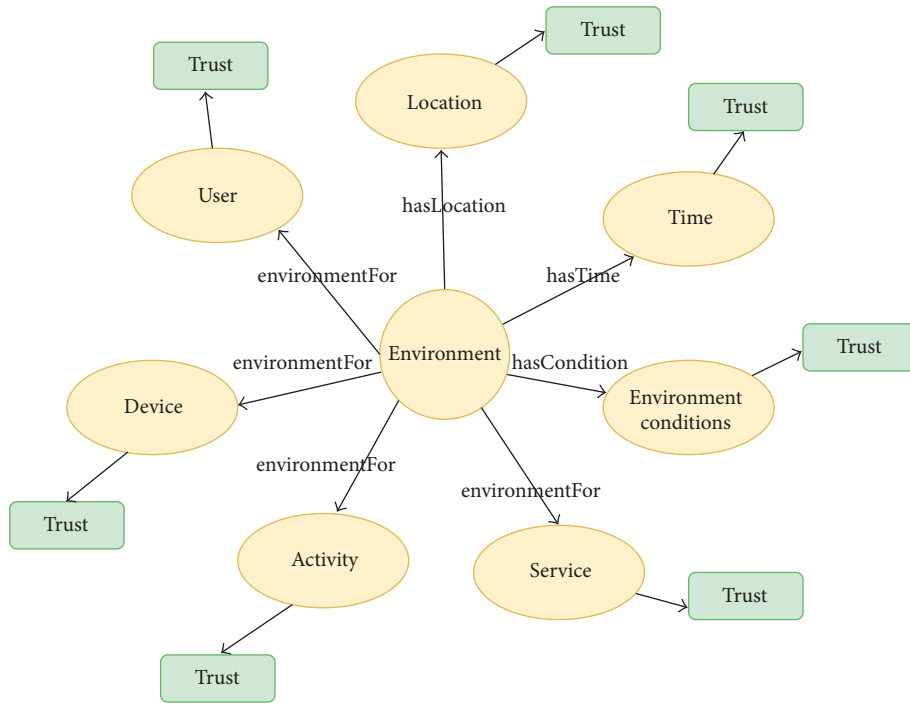FIGURE 10: Cognitive upper and lower domain context ontology model.



FIGURE 11: Relationship between the framework ontology concepts and trust.

device. It describes such aspects as the network maximum bandwidth, network minimum guaranteed bandwidth, error rate, network delay formation, and currently available network bandwidth.

These characteristics can be used either to enhance a minimum level of quality of media consumption or even to help to decide between different versions of the same media item.

*Service*: information about candidate services, for example, multimedia content such as streaming movies, to recommend is also an important concept. The most important features of the class are the metadata and the presentation format of the service. From this concept, let us

take a streaming movie as an example, from which information about the genre, titles, and so on can be obtained. This information could be used in the recommendation process to filter important characteristics of the audiovisual content, such as video-audio formats and so on.

*Adaptation*: the adaptation concept describes the presentation of the service items based on the device characteristics, environment conditions, network conditions, and so on [20, 35]. The adaptation concept relates to other concepts such as the environment, device, network, time, and so on. The concept can be used to define rules to determine when and how a service should be delivered. For example, if
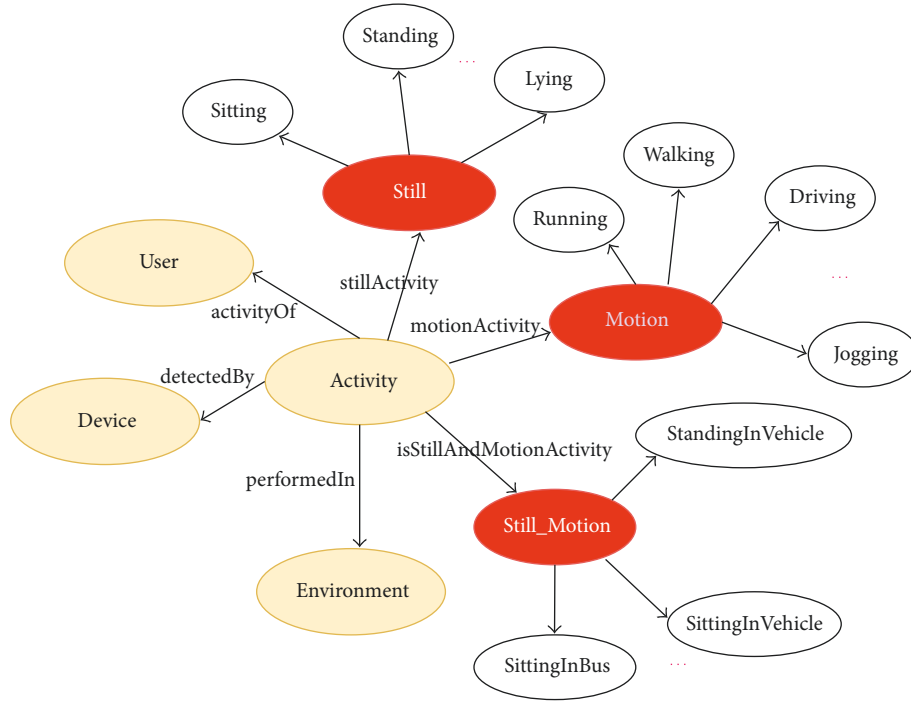
FIGURE 12: Activity concept [40].

the network condition such as bandwidth is very poor while delivering a streaming movie in HD, the adaptation decision could decide to deliver the content in formats that consume less computational resources.

*Trust*: trust has been used as a measure of confidence that an entity will behave in an expected manner despite the lack of the capability to monitor or control the environment in which it operates [5]. Trust is an important concept as we enforce confidence in items being suggested to the target users. In our framework, trust is used to enforce confidence of the target users in the recommendations that are provided in the given contexts. In the semantic model, trust is used as a property of other concepts (Figure 11), whereby every concept has an associated trust property and it is measured based on the feedback obtained from users after consuming services in specific contextual situation.

*(2) Cognitive context reasoning using SWRL.* The knowledge-based or cognitive context model incorporates reasoning mechanisms based on two methods. The first method is the inherent ontology reasoning mechanism that is responsible for checking class/concept consistency and implied relationships. This method uses the inference engine such as Pellet or Jena to provide functionality for checking the consistency of ontologies, computing the classification hierarchy, explaining inferences, and answering queries [41].

The second method is based on the Semantic Web Rule Language (SWRL) [42, 43]. Because OWL does not provide the mechanisms for expressing all relations between concepts in the ontology model, OWL has been extended with SWRL to allow inferring new knowledge from multiple facts or conditions at the same time by providing mechanisms for expressing complex relations. For example, OWL cannot express the relation between a child and married parents

because it does not have the mechanism to express the relation between individuals with which an individual has relations. For instance, OWL cannot relate that a user, say Ana, is the child of the married parents James and Comfort. Nevertheless, with SWRL, this can be expressed in the following way: $user\,(?u)^{\wedge}hasParent\,(?u,?f)^{\wedge}hasSpouse\,(?f,?m)\rightarrow childOfMarriedParents\,(?u)$. This specification adds rules to OWL ontologies, while also providing an extra layer of expressivity and the capability to infer additional information from an OWL knowledge base. Basically, SWRL consists of antecedent and consequent parts, which in turn consist of positive connections of atoms. Informally, SWRL can provide high-level reasoning in the following form: *if all atoms in the antecedent are true, then all the consequents of the rule must be true or vice versa*. Besides, SWRL comes with built-in unary predicates for describing relations between classes and data types, binary predicates for data properties, and some *n*-ary predicates such as arithmetic operators to compute desired behaviors. For example, it is easier using SWRL to infer if Ana, in the previous example, is an adult or not using these built-in predicates as follows: $user\,(?u)^{\wedge}hasAge\,(?u,?age)^{\wedge}swrl: greaterThan\,(?age,18)\rightarrow adult\,(?u)$. Likewise, by combining time and location data, the type of the device, and whether the user is accompanied or not, as well as user activity, the system will be able to infer the situational contexts of the user, for example, *Ana is sitting at home*.

In listings 1–3 of Figure 13, we provide some examples of specific domain rules supported by the context ontology model. Listing 1 is an example that infers that the user is indoor, sitting at home, and on a weekend. In listing 2, the user is in outdoor in the office walking, whereas in listing 3, the user is indoor, sitting in the office. Using this contextual information, personalization systems can be designed to

Listing 1
*Context:hasUserId (?user, ?userId) ∧context:atHome (?user, ?location) ∧ context:weekEndAtLocation(?location, ?weekend) ∧*
*Context:locationTimeIs (?location, ?time) ∧context:isMorning(?location, ?time) ∧ context:inIndoorLocation (?user, "true") ∧*
*Context:hasUserActivity (?derived, ?activity)→context:hasHomeLocationActivity (?user, ?sitting)*
Listing 2
*Context:hasUserId (?user, ?userId) ∧context:atOffice (?user, ?location) ∧ context:weekDayAtLocation (?location, ?weekday) ∧*
*Context:locationTimeIs (?location, ?time) ∧ context:isEvening(?location, ?time) ∧ context:inOutdoorLocation (?user, "true") ∧*
*Context:hasUserActivity (?derived, ?activity)→context:hasOfficeLocationActivity (?user, ?walking)*
Listing 3
*Context:hasUserId (?user, ?userId) ∧context:atOffice (?user, ?location) ∧ context:weekDayAtLocation (?location, ?weekday) ∧*
*Context:locationTimeIs (?location, ?time) ∧ context:isEvening(?location, ?time) ∧ context:inOutdoorLocation (?user, "true") ∧*
*Context:hasUserActivity (?derived, ?activity)→context:hasOfficeLocationActivity (?user, ?sitting)*

FIGURE 13: Listings 1, 2, and 3: example cognitive rules for inferring higher-level contextual information using SWRL.

recommend relevant service items that suit these specific contextual situations and preferences of the user [40].

### 3.3. Trust Evaluation.

The *Trust Evaluation* aggregates trust-related information and feedback obtained from users whenever they consume services in specific contexts to derive an evaluation for the trustworthiness of context data. Such evaluation for the trustworthiness supports decision-making of users. The trust evaluation is done using the following key modules as illustrated in Figure 14:

(i) *Trust Data Collection and Preprocessing*: the data collection implements the trust data collection agents using the available interfaces or APIs provided by the IoT platform data access layer such as the FIWARE [44] context broker or other REST interfaces, to collect or gather trust-related data. The data could be opinions of entities (users) as feedbacks on services consumed and so on.

(ii) *Trust Feature Computation*: the feature computation is responsible for computing/extracting features from the collected trust-related data. It also preprocesses the collected data to eliminate erroneous or repetitive and other irrelevant data for trust indicators' evaluation.

(iii) *Trust Data Store*: this module is responsible for storing trust-related data and the history of interactions between entities of the IoT platform. Additionally, this historical data store stores trust scores for the entities of the platform. Any trust score consumers, for example, recommendation manager, can interact with the trust data store via a web service interface, or via the Wise-IoT data access component such as the FIWARE broker to request such data.

(iv) *Trust Indicator Computation*: trust indicator computation implements one of the functionalities of the trust evaluation, realized based on the REK model [16]. It is responsible for calculating trust indicators such as experience, reputation, and knowledge.

(v) *Trust Score Evaluator*: this implements the REK model, the main module responsible for evaluating and generating trust scores for IoT entities of the framework. The trust evaluator computes the trust score or index from the combination of trust indicators, for example, experience, reputation, and knowledge.

In summary, the trust evaluation module subscribes to the IoT context broker. Whenever a user submits a feedback about the service he/she has just consumed, the feedback manager sends the feedback information to the trust component (TC), using the provided API. Details of mechanisms for trust score computation can be found in [16].

### 3.4. Contextual Entity Profiling.

We define a user as any entity that has the capability to provide and consume services be it data or any other information service. In this section, we present a profile/preference model capable of using the user's contextual situation, preferences, and other related information to personalize and deliver relevant services.

### 3.4.1. Contextual User Preference Definition.

The user profile describes preferences of a user in the form of a summary, normally based on the history of the user's actions such as consumption or provisioning of services [9]. The contextual user profile service (CUPS), based on our previous work in [11, 20], summarizes the user's service consumptions into a limited set of categories. Categories can be characterized by one or more services, and such services can be characterized by many properties or attributes. Several services can be associated with one or more service categories. In addition, the profiling model incorporates contextual dimension, associating one or more inferred context with each *category-service-property* relationship, with each context having an associated trustworthiness score. Thus, we have *category-service-property-context-trust concept*, presenting each user profile as a five-level tree, as shown in Figure 15, and the root of the tree represents the entity or user's optional information such as entity ids, timestamp, and so on. The first level of nodes corresponds to the service category; the second level represents the services associated with the specific service category; the third level contains the attributes of a given category-service. This level provides the service item's context, characterizing at a finer detail, the consumed service, and thus the preferences of the user. A limited set of properties is used for each service to obtain a good compromise between sufficient degree of characterization of service (hence, sufficient ability to make distinctions) and reasonable dimensions of the
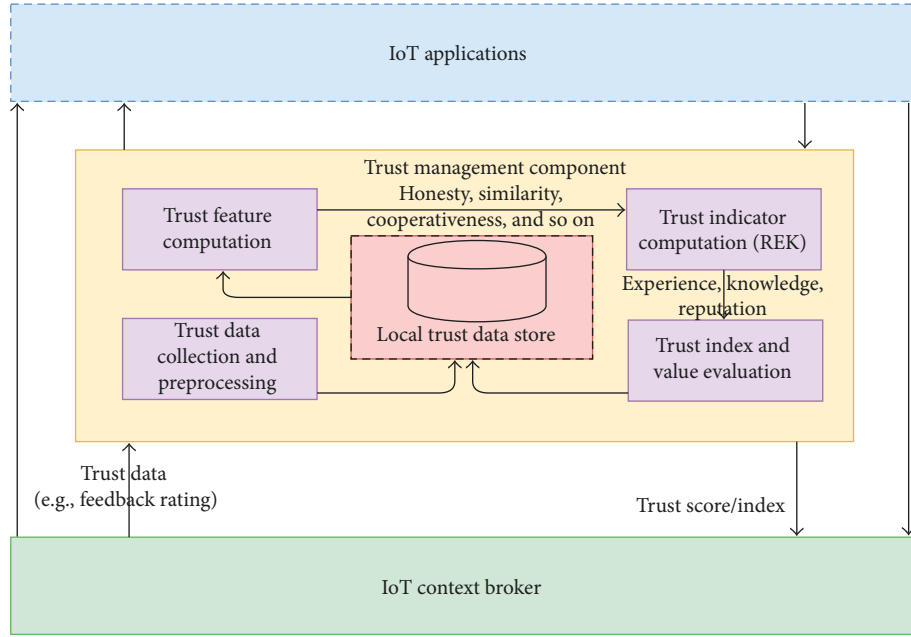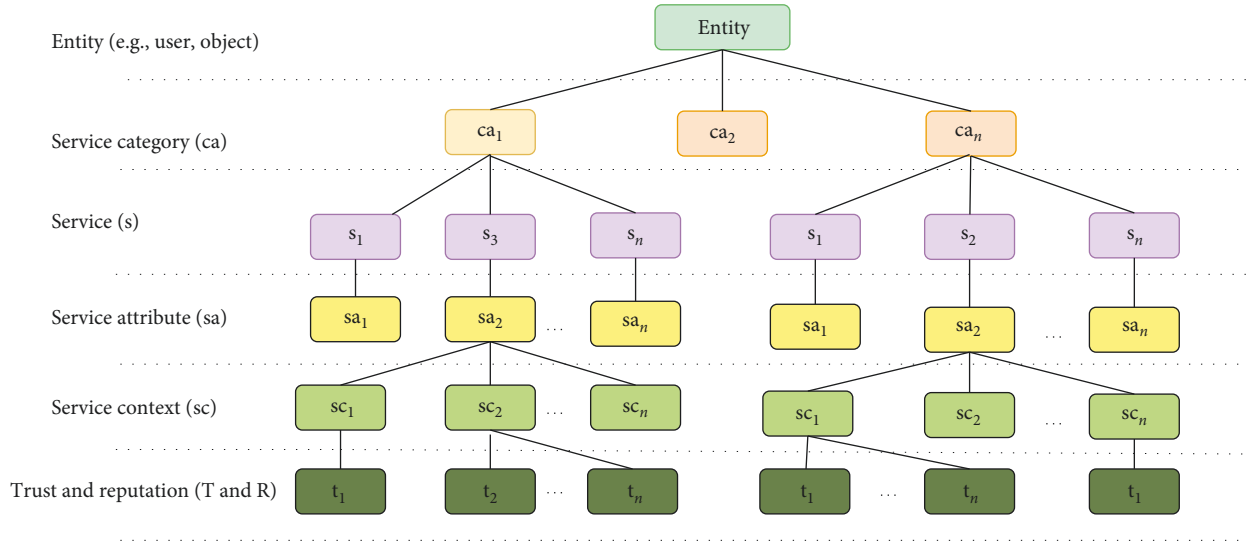
FIGURE 14: Trust computation component.



FIGURE 15: The preference model, showing the structure and relationships between entities in the profile model.

user profile. The leaf nodes provide information about the contexts (where the user preferences have been expressed) and associated trust score representing trustworthiness of the context information. Leaf nodes have four numerical fields (type, trust score, intensity, and *lifetime*), whereas all other nodes have only the type field. In the leaves, the types represent the type of context. The introduced concepts in the user profile, the intensity, trust score (as an extension), and lifetime, track the user's contextual consumption history, allowing incorporation of the trustworthiness of the contextual situation wherein the users consume services.

Using these weighted parameters, the system can determine the services that are important and relevant to the target user based on his/her contextual preferences with some measure of confidence. The *intensity* provides information about the number of times the user has consumed items of that *category-service-attribute* in that specific context. The intensity (the dynamic preference of the user) of those elements belonging to the item's content tis calculated by summing up the products (weight × lifetime) of all their occurrences. The intensity value of the retained elements at this level is obtained by visiting their child nodes in a breadth-first traversal. The same applies to the retained elements of the category level. The intensity of the elements that belong to the genre level is the largest value of their children. Accordingly, these values are obtained by performing a depth-first traversal. This way, the user profile can handle any category of services, for example, services

providing multimedia items such as movie, news, and music to name a few.

The intensity (the dynamic preference of the user) is computed by summing up the product (weight × lifetime × trust score), given in (4), of all their occurrences.

$$P_{ri} = \sum_{i=1}^{n} (\text{weight} \times \text{lifetime} \times \text{trust}) \qquad (1)$$

*3.4.2. Entity Profile Definition.* Let $E$ be the set of $N$ things or entities, and entities can be humans (users), objects, or devices. Each entity $E = \{e_1, e_2, \ldots, e_n\}$ is defined by $m$ set of optional demographic attributes $A = \{A_1, A_2, A_3, \ldots, A_m\}$ and a set of preferences $\Pr(u_i) = \{pr_1, pr_2, pr_3, \ldots, pr_{ir}\}$, where each preference $\Pr_{im}$ of user a is further defined by a set of attributes $SA = \{sa_1, sa_2, sa_3, \ldots, sa_p\}$. Furthermore, we have a set of high-level context information $SC = \{sc_1, sc_2, sc_3, \ldots, sc_n\}$ associated with each preference $\Pr_i$.

In the above definitions, context SC has a very complex structure, which reflects the nature of the difficulties in accurately representing context information. However, in this paper, it is assumed that the context information is defined by a structure that allows atomic contexts to be related in a way that allows high-level contextual information to be inferred, that is, a three-layer structure of inferring from low-level sensor data, a high-level atomic context information, which can be related to another context information to arrive at a higher-level semantic context information. For instance, from the above definition, an entity $e_i$ where $e_i \in E = \{e_1, e_2, e_3, \ldots, e_n\}$ is defined by a set of demographic information $A = \{$entityId, Name, etc.$\}$, by a set of preferences $\Pr = \{pr_1, pr_2, pr_3, \ldots, pr_n\}$. Each service is classified into a category $Ca = \{cr_1, cr_2, cr_3, \ldots, cr_n\}$. Each category is characterized by a set of services $S = \{s_1, s_2, s_3, \ldots, s_n\}$, and each service is characterized by a set of attributes $Sa = \{sa_1, sa_2, sa_3, \ldots, sa_n\}$. In addition, each context $sc_i$ defined above is associated with the set $Tr = \{tr_1, \ldots, tr_n\}$ of trust. The trust value defines the confidence of the user in services recommended to his/her in that context. The model of the user preference can then be built in a general form as follows:

$Y = f(X_1, X_2, X_3, \ldots, X_n)$, where $X_1, X_2, X_3, \ldots, X_n$ represent the profile information such as the set of service attributes, $A$, and preferences, Pr, characterized by the sets $S$, Ca, and SC with associated trust values Tr. $Y$ is the dependent variable (representing user preference) to be predicted, and function $f$ is the predictive function as provided in Section 3.4.3. For example, we can predict the services that user $u_i$ will like to consume using this model. This model can be defined such that we can have context-based user preferences, context-aware trust-based user preferences, and user preferences without context and trust information as defined in (2)–(4), respectively.

$$Y_{ct} = f_{ct}(X_1 C_1 T_1, X_2 C_2 T_2, \ldots, X_n C_n T_n), \qquad (2)$$

$$Y_c = f_c(X_1 C_1, X_2 C_2, X_3 C_3, \ldots, X_n C_n), \qquad (3)$$

$$Y = f(X_1, X_2, X_3, \ldots, X_n). \qquad (4)$$

These models can be used to build the contextual user profile and can also be switched to build the traditional user profile.

The parameters $c_{i=1}, \ldots, c_{i=n}$ represent the contextual information $t_{i=1}, \ldots, t_{i=n}$; $f_c, f_{ct}$, and $f$ are the functions for retrieving users' contextual preferences, user context-aware trust-based preferences, and ordinary user preferences, respectively. As (2)–(4) illustrate, the profile model can be used to learn the user profile in three modes. It works in the contextual mode and in the noncontextual or traditional mode to determine the user preferences with and without using contextual information, respectively. The traditional mode (non-contextual mode) is executed in situations where the user's contextual information is not available or when it is difficult or impossible to acquire. In this mode, an entire profile considering the user consumption history (if any) is used. Alternatively, it could use the consumption history of users who have similar preferences to the current user to learn the target user's preference. The default mode is the contextual mode in which the user's contextual information is used to learn the user's preferences. The contextual mode allows us to incorporate trust information of the context information, and thus, the ability to evaluate the preferences is not only based on contexts but also based on the trust information of the contexts.

*3.4.3. Feedback Mechanism.* To learn the user's service consumption preferences based on the entity profile model in the last section, the proposed system adopts the relevance feedback method, which we have used in our previous solution [11]. Relevance feedback is a technique used to obtain a user's opinion on recommendations. It can speed up the user preference learning process and improve the quality of recommendations [42]. There are two methods in literature used for learning user preferences based on user feedback, namely, implicit and explicit relevance feedback [9]. In implicit user feedback, the system observes the content consumptions by the user and records information about the consumed content. In our case, such information includes category, genre, property of the item, and most importantly the context in which users consume this item. The user feedback process, without asking for any information from the user, assigns a relevance value to the consumed content using the content metadata information as well as the context of consumption. The explicit user feedback involves asking the user to provide ratings or some form of evaluation of the relevance of the consumed content, using metrics such as like or dislike of the recommended items. In mobile environments, this approach can be obtrusive as it distracts and consequently bores users, discouraging them from using the recommendation system, which consistently requires them to provide rating information every time they consume content [12]. For this reason, we adopted an implicit user feedback method in combination with a simpler form of explicit user feedback. Our approach neither asks users for explicit rating

information nor measures how long a user has spent during the consumption of any service, for example. Measuring time spent on recommended items as argued by Bjelica [42] may lead to wrong conclusions. Thus, rather than measuring time spent when consuming content, we used a combination of the contextual user profile learning model and the context in which users respond to recommendation either by clicking or not (e.g., the device's screen) to learn the user's feedback on the recommended items. It then extracts implicit information about such items to learn whether users like or dislike such items by assigning what we call relevance values to the extracted information, such as category of the content and its genre, taking into consideration the contextual situation of the user at that given time. The system automatically assigns these relevance values in two numeric formats as illustrated in (5)–(7) consisting of *weight w* and *lifetime γ* parameters. Equation (5) represents positive feedback, whereas (6) represents negative feedback. Equation (7) is used as a decay function, which updates the preferences as the elapses.

An intensity $w_{ij}$ represents the relevance of a service with the preference $\mathrm{pr}_i$ belonging to the contextual situation $C$ that entity $e_i$ consumes in context $C_i$. Whenever the context awareness model detects that the entity is in a context $C_i$, for a continuous period of time $[0, T]$, the reasoner finds the preference of service $s_1, s_2, s_3, \ldots, s_n$, whose contextual situations match the present contextual situation of the entity; then, weights $w_i c_i \in [0, 1]$ are associated with this service preference, which at time $T$ are updated as follows:

$$w_i c_i = w_{i-1} c_{i-1} + \gamma \left( \alpha - w_{i-1} c_{i-1} \right), \quad i = 1, 2, 3, \ldots, n. \quad (5)$$

Then, for those services $b_1, b_2, b_3, \ldots, b_n$ with associated weights $w_1 b_1, w_2 b_2, w_3 b_3, \ldots, w_n b_n$ not consumed by the user, these weights are updated as follows:

$$w_i c_i = w_{i-1} c_{i-1} - \gamma \left( \alpha - w_{i-1} c_{i-1} \right), \quad i = 1, 2, 3, \ldots, n, \quad (6)$$

where $\gamma$ is a learning parameter whose value is obtained by

$$\gamma = 1 - \left( \frac{t}{45} \right)^5. \quad (7)$$

$\alpha \in [0, 1]$ and its value is set to 1 in (5) and 0 in (6).

Equation (7) was derived so that certain attributes in the user profile that represent items that have not been seen for a long period (note that we have considered the long period to be 30 days) will eventually have no impact on the preference evaluation. However, unlike our previous solution in [11], where the weight $w_i$ provides information on the number of times the user has consumed items in specific contexts, here $w_i$ represents the trust $\mathrm{tr}_i$ of context of consumption computed from feedback information obtained from users. The lifetime parameter, $\gamma$, provides an indication of the time elapsed since the last consumption occurred. Its value is set to 1 when the user consumes a service that belongs to a *category-service-attribute* of Figure 14 and periodically decrements it if the consumed service does not belong to such category-service-attribute. Smaller values indicate that the user has lost interest in that type of service, regardless of the value of its weight or contexts. We also use this value to determine if a trustworthiness can be considered valid or not.

For example, in practice, it allows to give more importance to contexts of items consumed recently and less to those consumed long time ago. The factor $\alpha$ has a value in the range $[0, 1]$, assigning less or more importance to new category-service-attribute. For newly created category-service-attribute, $\alpha$ assumes the value of 1; otherwise its value, determined based on experiments, is set at 0.5. The value of $-1$ is used to indicate that the user has rejected or not preferred a service item with those properties.

The *lifetime* parameter as explained earlier is set to 1 for matching items in the user profile, by assigning the value 0 to the factor $t$. The factor $t$ represents the number of days elapsed since the last time the user has consumed an item with the characteristics described by the profile. With (7), the relative importance of the *category-service-property* of an item consumed by a user remains above 0.9 in the first 30 days after it has been visited, rapidly decreasing to zero after that period (nonnegative values are automatically converted to zero). For all other *category-service-attribute* in the user profile, the update of the *lifetime* parameter is performed by linearly increasing the value of $t$. Thus, category-service-attribute of a service that has been consumed before but has not been seen or consumed for a long period will have either low or no impact on the user preference evaluation. Also, note that the trust score influences the relevance of a given service.

*3.5. Incorporating Trust as an Extension of the Context-Aware Recommendation Process.* In this section, we demonstrate how trust can be incorporated into context-aware collaborative recommendations. The process of incorporating trust into the recommendation system, especially context-aware collaborative recommendations, is in three key important steps, namely, building entity profile, generating entity trust, and predicting service preferences for the target user.

In recommendation systems, especially collaborative recommender systems, opinions of the neighbors (or so-called friends) of a target user are used to suggest interesting items. Thus, it is required that such neighbors whose opinions are used to compute suggestions are those neighbors that are trusted by the target user. In a social network environment, people are usually influenced by the opinions of friends with whom they have established trust relationships. Trust-based recommendations will enhance the acceptability of recommendations provided by the system when it uses information from trusted sources. Therefore, incorporating trust into recommendation processes as shown in Figure 6 will not only improve the accuracy of the recommendations but also has the potentials to enhance the user experience. Service consumers will only receive recommendations from those who are in their trust networks. With trust, it is also possible to address the challenges of the traditional recommender systems [15]. However, there is no clear explanation on how it can be used in CARSs. We proposed to incorporate trust into CARSs with three main components as we have earlier illustrated in Figure 2, namely, trust-related information or trust scores, context information, and recommendation mechanisms (including contextual preferences) as analyzed in the previous sections.

*3.5.1. Building Entity Profile.* Generally, this process involves associating user preferences with contexts in which such preferences have been expressed. As discussed in Section 3.3, every time a user consumes a service, information such as the context of consumption, relevance of that service in such a context, which is given as rating provided by the user after consuming the service or through an implicit means, and the computed trust score for the given context is captured in the user profile. Also, the user profile is updated using a relevance feedback mechanism explained in Section 3.4.3 Thus, the context, the preference, the relevance feedback, and other information about the consumed service will make up the entity or user profile.

*3.5.2. Trust Score Computation.* The second step, which involves how the trust score is computed for any entity such as the context of a user, is beyond the scope of the current paper. However, Jayasinghe et al. have provided computational models for trust computation in [16], interested readers can check the reference for details. However, the trust score for a given context of consumption is computed using the implicit or explicit rating information obtained from the relevance feedback process, and this score is associated with the context, service, and preference of the service for a given user in every consumption session.

*3.5.3. Generating Predicted Preference.* In this section, we illustrate how trust can be used to generate predicted preference for a given item or service for a target user.

In a typical collaborative recommendation system, the goal is to predict rating (preference) of the target users based on the similarity of other users with similar taste. This concept has been extended in context-aware collaborative recommendation where the goal is to first determine users who prefer certain items in a similar context and then exploit this information to predict the preferences of the target user in such contexts [11]. Thus, in a traditional collaborative recommendation, typically the Pearson Correlation Coefficient (PCC) is used to find the degree of linear relationships between two entities. In this article, we modified the PCC to incorporate contexts. This means that the degree of linear relationships between two entities takes their contexts into account. Thus, the similarity between an entity $v$ and its neighbor $u$ in context $c$ can be defined by the PCC equation as follows:

$$w_{v,u,c} = \frac{\sum_{i=1}^{m} \left( p_{v,i,x} - \overline{p_v} \right)\left( p_{u,i,c} - \overline{p_u} \right)}{\sigma_v \sigma_u}. \tag{8}$$

Using this equation, the CARSs can combine the similarity between the relative preferences of users $v$ and $u$ for the same item in context $c$ for all the items they have both consumed and preferred in that context or other similar contexts. Usually, it generates one of the two values: +1 or −1. +1 means that users $v$ and $u$ have similar taste, and −1 means that they have dissimilar taste.

After obtaining the similarity between $v$ and $u$, CARSs process the profiles of all friends of $v$, that is, those users who have similar taste as the target user by computing the average of their previous preferences in the rating form, using the PCC value generated above as a weight [25]. The predicted preference for the target user for the item $i$ can then be computed using

$$\mathrm{pr}_{v,i,c} = \overline{p_v} + j \sum_{u=i}^{n} \left( p_{u,i} - \overline{p_u} \right) w_{v,u}. \tag{9}$$

To introduce trust into the context-aware collaborative model, each contextual preference is associated with at least a context and a trust score as explained in Section 3.3. Now, we define a contextual trust-based preference as follows:

$$P_{u,i,c,t} = j \sum_{x \in C} \sum_{i=1}^{m} \left( p_{u,i,c} \right) \mathrm{sim}_t(c, x) T_c, \tag{10}$$

where $\mathrm{sim}_t(c, x)$ is the similarity between the target user's current context and context of his/her friends. $T_c$ represents the trust scores for the contexts of those friends. The inner sum loops over each context dimension and the inner sum loops over all context trust scores in that dimension to obtain their trust scores. Interested readers can see [11] where we presented a model for computing similarity between two contexts.

We now substitute $P_{u,i,c,t}$ in (8) to have

$$\mathrm{pr}_{v,i} = \overline{p_v} + j \sum_{u=i}^{n} \left( P_{u,i,c,t} - \overline{p_u} \right) w_{v,u}. \tag{11}$$

This model combines all weighted preferences with respect to similarity between the target user's context, contexts of his/her friends, and their corresponding trust scores to provide an overall context-aware trust-based predicted preference for the target user in the current context. In Section 4, we would be evaluating the impact of context and trust on the relevance of recommendations.

## 4. Experimental Evaluation and Results

In this section, we provide our initial experimental evaluations of the proposed conceptual framework based on the context-aware preference models. But first, we define the metrics used in the evaluations and then we provide details of the evaluations.

### 4.1. Evaluation Metrics

*4.1.1. Precision.* Precision can be broadly defined as the proportions of the top $n$ recommended items that are relevant. It is the number of relevant items selected from recommended items to the number of items that are relevant in the recommended set. It represents the probability that a selected recommended item is relevant according to the preferences of the user.

In context-aware recommendation systems, precision represents the probability that the selected items among

those recommended in the current context are relevant in such context as defined in (12):

$$P = \frac{N_{rc}}{N_c}, \tag{12}$$

where $N_{rc}$ is the number of items from the recommendation list selected by the users as relevant in the context of use. $N_c$ is the number of relevant items.

### 4.1.2. Recall.
Recall is the ratio of the recommended media items relevant and preferred by the users in the current context (for contextual recommendations) to the total number of relevant items in the recommendation set as defined in (13). In other words, it measures the proportions of all relevant items included in the top $n$ ranked recommended items.

$$R = \frac{N_{rc}}{N_r}, \tag{13}$$

where $N_r$ is the number of selected relevant items in the recommendation set.

### 4.1.3. F-score.
F-score is the harmonic mean or weighted average of precision and recall. It is a measure of the accuracy of the recommendation system in consideration of relevant and nonrelevant items included in the recommendation set. In practice, precision and recall are inversely proportional and are both dependent on the number of recommended items returned to the user. When more items are returned, recall tends to increase whilst precision decreases and vice versa. Thus, evaluating the detailed accuracy of the performance of a recommendation system becomes difficult using only precisions and recalls individually. However, both metrics can be combined using the standard F-score metric to evaluate the quality of a recommendation system to address this conflict as shown in

$$F\text{-score} = \frac{2PR}{P + R}. \tag{14}$$

An example of how to compute these metrics is given in Figure 16, and the precision, recall, and F-score of item number 5 in the ordered recommendation list are obtained as follows.

From the recommendation list, which contains ten items, only seven are relevant.

$N_r = 7$, $N_{rc} = 4$, $N_c = 5$, $P = 4/5 = 0.8$, $R = 4/7 = 0.57$, and F-score $= ((2 \times 0.8 \times 0.57)/(0.8 + 0.57)) = 0.67$.

### 4.1.4. Average Precision (AP@K).
Average precision (AP@K) is the mean precision score obtained for each relevant item at top $k$ recommendations in the test cases for every test profile computed as follows:

$$AP@K = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{k_i}, \tag{15}$$

where $m$ is the number of relevant items and $1/k_i = 0$, if item $i$ in the set is not relevant.

### 4.2. Evaluation Data.
Currently, no available open data suit and address the methods in this article. Thus, to evaluate the feasibility of the proposed framework, we conducted preliminary evaluations by using our existing anonymous survey data solicited from online users capturing movie consumption preferences of various users on mobile devices [11, 20, 40]. With these data, we created 200 unique user profiles, each with 19 different entries at the attribute level (the entry in the category level was the same for all users—movie-streaming service). High-level contexts, as presented in Section 3.2, were associated to these entries. Examples of such high-level contexts are provided in Table 1. The anonymous users were asked to associate these terms with location, time, and activity in which they consume any content with such terms. Using these data, we instantiated those contextual profiles representing contextual preferences. For the candidate movie items, online movie databases were explored. Thus, specific movie data were crawled from two popular online movie databases, namely, the IMDB and The Movie Database (TMDB). The crawled candidate content metadata for 5000 movie records from The Movie Database (themoviedb.org) further enhanced with additional metadata retrieved from the IMDB.

This metadata set consists of unique movie genres representing the service attributes in the preference model, and each record contains, on the average, 3 different genre labels. Genres were further characterized by language, cast, country, duration, and release date. These terms thus constitute the media item's attributes in the profile model. Given that users were anonymous and thus were not available to provide feedback, we devised an alternative approach to allow marking of recommended items as *relevant* or as *irrelevant* in an experimental context-aware movie streaming recommendation application we have developed [11]. This allowed to simulate the acceptance or rejection of those items by the user (Figure 17). Hence, an item is selected as relevant if two-thirds of the terms that appear in the item record (but not less than 2 terms) also appeared in the user profile with a weight larger than the average weight of all terms in the user profile. Specifically, a recommended movie item with a metadata record presenting 3 terms is marked as relevant if 2 out of these 3 terms appear in the user profile with weights larger than the average weight of all terms. Otherwise, it is marked as irrelevant. We adopted this approach because we observed that the number of terms contained in the item metadata profile or record along the *attribute-context* concept influenced the classification of candidate items.

### 4.3. User Profile-Centered Evaluation.
To understand the impact of context and trust on quality of recommendations according to the user preferences either with or without context and trust, we examine the three preference models. In this section, we evaluate the three preference models defined in Section 3.4.3. These models represent those preferences and service consumption that are characterized by context information, trust, and traditional user preferences that are not characterized by these additional properties.
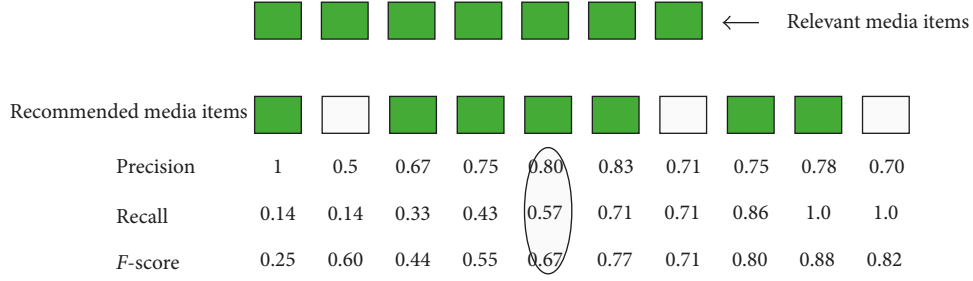
Precision       1      0.5    0.67   0.75   0.80   0.83   0.71   0.75   0.78   0.70

Recall        0.14   0.14   0.33   0.43   0.57   0.71   0.71   0.86   1.0    1.0

F-score       0.25   0.60   0.44   0.55   0.67   0.77   0.71   0.80   0.88   0.82

FIGURE 16: Computing precision, recall, and F-score for the recommendation system [40].

TABLE 1: Sample contextual information.

| User | High-level contexts |
|---|---|
| User 1 | {(DayOfWeek: Sunday), (TimeOfDay: Evening), (Activity: Sitting), (Location: Cinema), (Illumination: Bright), (Noise Level: Normal)} |
| | . . . |
| User 2 | {(DayOfWeek: Friday), (TimeOfDay: Evening), (Activity: Sitting), (Location: Home), (Illumination: Bright), (Noise Level: Normal)} |
| | . . . |
| User 3 | {(DayOfWeek: Monday), (TimeOfDay: Evening), (Activity: Jogging), (Location: Sport Complex), (Illumination: Bright), (Noise Level: Normal)} |
| | . . . |
| User 4 | {(DayOfWeek: Monday), (TimeOfDay: Evening), (Activity: Sitting), (Location: Office), (Illumination: Bright), (Noise Level: Normal)} |
| | . . . |
| User 5 . . . | {(DayOfWeek: Friday), (TimeOfDay: Evening), (Activity: Sitting), (Location: Home), (Illumination: Bright), (Noise Level: Normal)} |
| | . . . |

*4.3.1. Traditional Preference Evaluation.* In the first experiment, the user preferences were computed without considering the consumption contexts. Of course, this also means that we do not consider whether consumption context is trustworthy or not. The user preferences have been retrieved with no consideration for these additional characterizations by context and trust. Thus, recommendations were generated based only on the preference values. After analyzing the recommendations based on the content of user profiles and movie items, we computed the F-score for each recommendation accordingly. Note that, in all experiments, 5 rounds (R1-R5) of recommendations were generated. Also, note that $n@j[j = 5, 10, 15, 20, 25]$ represents the number of recommendations generated.

The rationale was to observe if the model could find more relevant recommendations from the corpus of candidate' items as we increased the number of items in the recommendation set. As explained in Section 4.2, each recommendation set is generated, and its F-score is computed. Figure 18 shows the F-score as a measure of the accuracy of recommendations. In this figure, as the number of recommended items increased, the F-score also increased, but from $n = 20$ and 25, the F-score begins to fall. However, the best accuracy was obtained between $n = 10$ and 15,
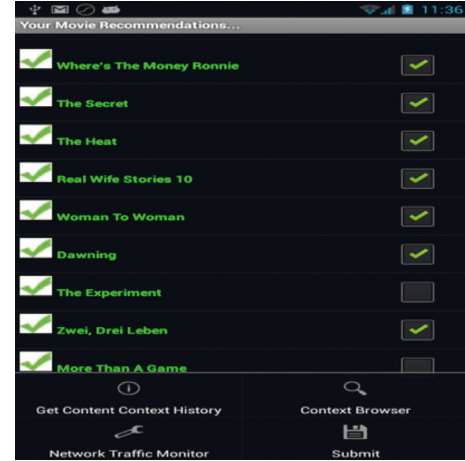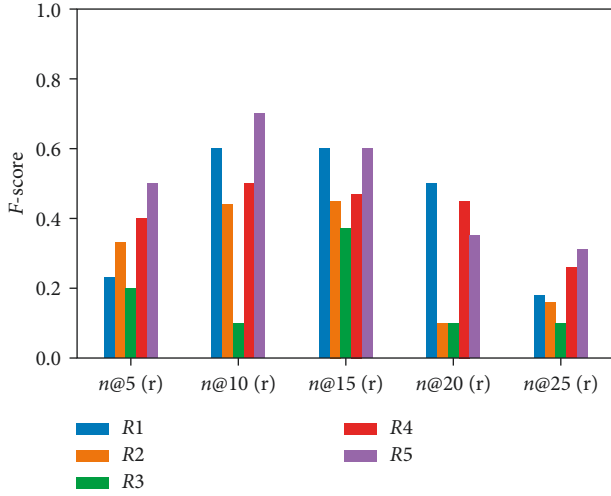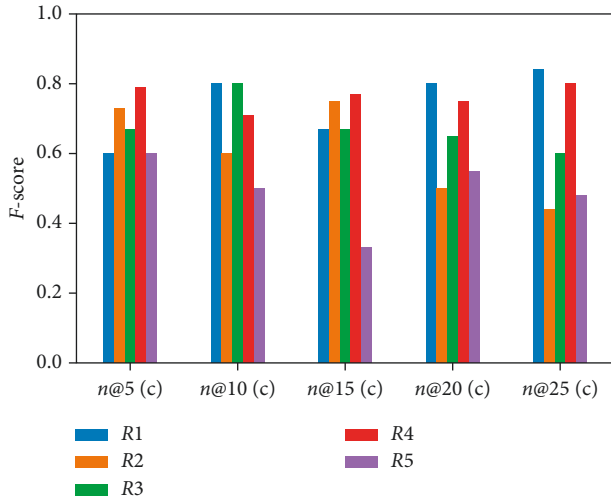


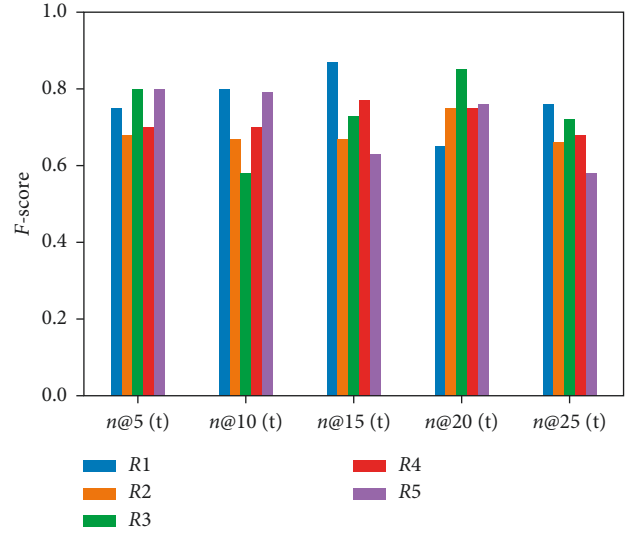FIGURE 17: Item relevance/irrelevance simulation interface on the mobile device.

getting up to 70%. The possible reason for this is that as the number of items in each set of recommendation increases, the relevant items left in the corpus also decrease, which means that it is difficult for the system to obtain relevant items as this number increases.

*4.3.2. Contextual User Preference Evaluation.* The second evaluation involves generating the user preferences with specific contexts, for example, movies in the category-service-attribute-context. In this case, the records of services (streaming movies in this instance) consumed by users in specific contexts being processed have been recommended according to the contextual situations of the users. In practice, this translates into setting average weights for preferences assigned to services and properties in specific contexts, whereas others are assigned very low weights or even zero by the system since the user might not prefer such a service in that specific contextual situation. This means that if a user consumes a service/item in each context, such preferences receive bigger weights than those services consumed where contexts of consumption were not considered. This preference model does not consider the trust scores for the context entities in the evaluation process. We call this scenario contextual profiling. The F-scores were computed as in the last experiments for the contextual profiles. Figure 19 illustrates the F-scores obtained for the context-based preference model. We observed an improvement in the F-scores

Figure 18: *F*-score for noncontextual preferences.



Figure 19: Comparison of *F*-score for contextual preferences.



Figure 20: Comparison of *F*-scores for preference with trust-based contextual preferences.

computed for each round of recommendations compared to the traditional preference model in the last section. The improvement can be explained as being the result of the capability of the system for using contexts to filter the candidate items according to the contextual preferences of the users. We also observed that the system maintains better *F*-score as the number of items in the recommendation set increases, sometimes up to 0.8.

*4.3.3. Context-Aware Trust-Based User Preference Evaluation.* In this section, we provide an experiment conducted to study the impact of context and trust on recommendation accuracy using *F*-score as an evaluation metric as we did in the previous sections. In the current evaluation, however, we evaluate the possible impacts of trust on recommendation quality. Other possible evaluations could be to determine the impact of using untrustworthy profiles to compute the preferences of the target users. We would leave that for the next phase of our project. As explained in Section 3.3, we extended and developed a dynamic context-based

preference model with the capability to incorporate trustworthiness of contexts in which preferences are expressed by any consumption entity (users). For each preference expressed, there exist associated contexts and trust scores. Trust can be used in the recommendation filtering process for eliminating or excluding preferences of any profile considered untrustworthy based on its associated trust score. In fact, the profile containing such profile can be entirely excluded from the recommendation process. The ground truth is that the possible consequence of this would be an improvement in the accuracy of recommendations.

Thus, the goal of the current evaluation is to study the impact of trust on CARSs. In the recommendation process, preferences whose contexts' trust scores are below the threshold, which we set at 0.5 [0, 1] being the range of the trust values, are excluded from the recommendation process. Using this mechanism, recommendations were generated as done in the previous experiments. *F*-score for each round of recommendations is computed, and the obtained results in comparison to the other preference models are as shown in Figure 20. The figure shows that using context and trust-related information in the recommendation filtering process can improve the recommendation quality. For example, average *F*-scores at $N = 5$ for the 3 models are 0.68, 0.25, and 0.74, respectively, showing a progressive improvement in the recommendation accuracy.

In addition to evaluating the proposed solution using *F*-score, we have also compared the performances of traditional, context-, and trust-based recommendations by computing the average precision as illustrated in (15) of Section 4.1. The results obtained are illustrated in Figure 21. The result follows similar trends in the other evaluations presented above.

*4.3.4. Comparison of Processing Time for Traditional, Context-, and Trust-Based Recommendations.* One of the key
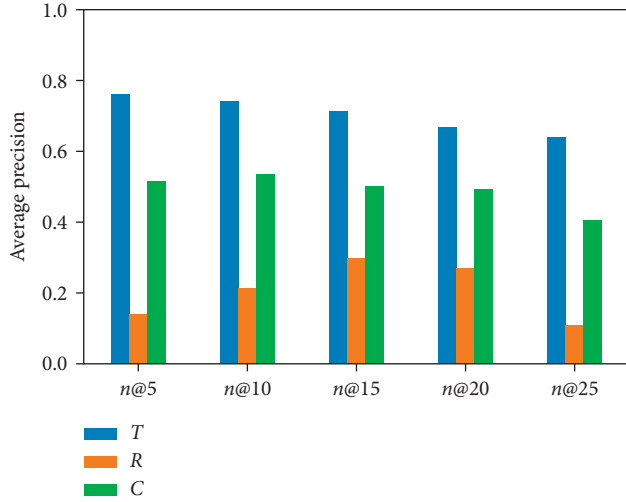
FIGURE 21: Comparison of average precision for traditional recommendation (*R*), context-based recommendation (*C*), and trust-based recommendation (*T*).



FIGURE 22: Comparison of time taken by traditional recommendation (*R*), context-based recommendation (*C*), and trust-based recommendation (*T*) to generate ranked recommendations.

performance metrics we have considered for measuring the quality of experience and thus of recommendation is the time taken to execute the recommendation execute and generate a set of ranked recommendations. In this regard, we computed the time taken to obtain recommendations with context and trust and without context and trust. The experimental system contains Intel® Core™ i7-3610 QM CPU @ 2.30 GHz, with 8 GB RAM, running 64-bit Windows 10 Pro. Like the previous experiments, recommendations were generated and ranked with 5, 10, 15, 20, and 25. Figure 22 shows the $n@j[j = 5, 10, 15, 20, 25]$ and the processing time. Generally, the computational time increases with the number of items in the recommendation sets. However, we observed that computation time for the trust-based process is higher compared to the context-based and the traditional processes. This is due to the additional computation required to determine the trust-related information from the user profile information.

## 5. Discussion

In this article, our goal is to present a conceptual framework including an architecture for context-aware personalization using cognitive contexts obtained from diverse sources including IoT devices and present how trust can be incorporated into the recommendation filtering process. The framework comprises components for realizing its specific functionality. From context sensing, context classification and inference to contextual entity preference modeling and recommendation mechanisms utilizing contexts, preferences, and trust to filter candidate services during the recommendation process. The work presented in this article is the initial stage of our work, which is based on developing a framework for context-aware trust-based personalized service delivery in IoT using mechanisms incorporating the elements of trust, context, and recommendation to personalize service delivery. The work, especially the trust
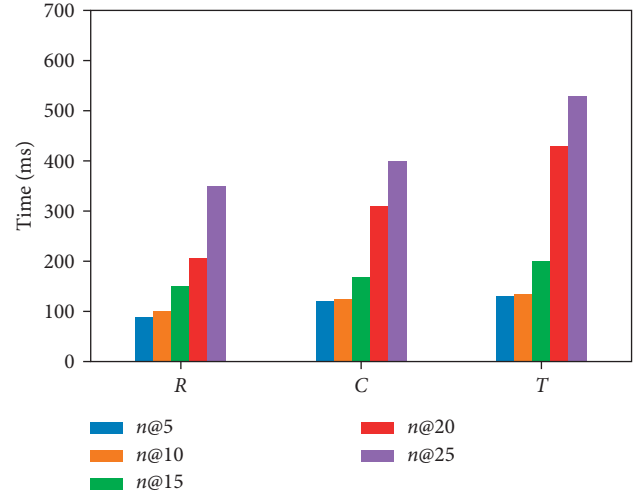
management component, is part of a European-Korean H2020 research project on interoperable global IoT [30].

The context awareness component of the framework was designed to utilize context sensing and classification mechanisms using machine learning algorithms, whereas the cognitive inference of higher-level or complex contextual situations of entities can be performed using semantic web technologies: ontological model, based on OWL, reasoning, and inference. The goal is to combine contexts from different sources including IoT objects to characterize the contextual situations of any entity in the ubiquitous environment.

To predict the preferences of the entity, a component has been proposed with the capability to learn such preferences in the contexts where they have been expressed. In many existing personalized systems, contexts of consumption have not been considered when filtering candidate items. And even those that utilize contexts of consumption do so with a limited set of static contexts, which are obtained directly from the users or from their mobile phones. The developed contextual preference model includes mechanisms to obtain feedback from users to adapt and improve the future personalization process. In addition, such feedback information can be used to compute and incorporate trustworthiness in the personalization process. The component can utilize the contextual preference model to filter and personalize candidate services using recommendation mechanisms. The goal is to use the contextual preferences and trust score associated with the consumption to filter such services. With this capability, we aim to improve the relevance of recommendations and improve user experience as our initial evaluation has confirmed.

To understand how contexts and trust can influence user preferences and consequently the quality of provided recommendations, we have performed some preliminary experiments to analyze the impacts of context and trust on recommendations. We evaluated the traditional preference model, which does not consider the contexts of consumption as well as the trustworthiness. We also evaluated the

contextual preference model. Finally, we evaluated the impact of trust on context-aware personalized recommendations as presented in the last section. The initial results show that traditional preference model performed poorly in terms of recommendation accuracy measured by *F*-score as a statistical test of the accuracy of recommendations. Further evaluation of the impact of contexts on the user preferences shows that more relevant recommendations can be provided when the consumption context is used in the filtering process. This result aligns with previous work where trust has been explored in traditional collaborative recommendations [15]. Further, incorporating trust into contexts of consumption can further improve the accuracy of recommendations. However, this comes with an additional computational cost in terms of processing time.

## 6. Conclusion

In this article, we have proposed a conceptual framework for exploiting the IoT's context awareness for predicting the user's preferences for personalized services. The article has discussed and elaborated the components of the proposed context-aware framework addressing some important functionality. First, the design addresses fourfold requirements, namely, (1) it proposes a context-aware framework that can collect, analyze, and infer high-level context information from IoT objects. (2) It can provide suggestions to users based on the context information, taking into account the contextual preferences of the users. (3) It proposes a context-based entity profile model for managing preferences. (4) It also proposes the incorporation of trust into context-aware personalized recommendations. Secondly, an integrated layered architecture of the system was proposed, and each component of the architecture has been elaborated based on how it realizes the system's functionality. One of the salient features of the framework is the contextual profile model with the capability to incorporate trustworthiness of contextual information. The proposed conceptual framework thus provides the following important features supporting context-aware personalization: context sensing, recognition, cognitive reasoning, and inference and modeling supports that provide a generic context awareness framework for mobile service personalization to address the problem of static contextualization of existing approaches. Contextual entity profiling supports the automatic learning of the entity's service consumptions in contexts, thereby allowing dynamic determination of its contextual preferences.

To validate our proposal, we conducted preliminary context-aware preference-centered evaluations of the profile models to establish the importance of contextual user preferences for filtering candidate service items in recommendation processes. We have also evaluated the importance of trustworthiness for context-aware personalization during the recommendation process. We have presented and discussed some initial results of the experimental validation of the proposed system. The results obtained emphasize the importance of these parameters for improving the quality of recommendations.

One key advantage of the proposed system is its unique ability that allows incorporating various context-aware recommendation algorithms, which can be implemented as components of the framework to deliver different kinds of recommendations.

Although the system shows some promise in terms of its potentials to improve the accuracy of recommendations and their relevance to the user's contextual preferences, we have evaluated only recommendations based on an extended collaborative filtering approach. We would like to experiment with other recommendation algorithms, especially content-based or hybrid-based recommendations incorporating context and trust. In addition, trust management mechanisms have been used in some existing works to provide security, reliability, and dependability of entities among others. However, we have not used trust in the current work to enforce security, but it has been used as a means to improving the quality of recommendations. Our future goal is to use trust as a means to ensuring that data or information from malicious entities is not allowed in the context-aware personalized service recommendation process. Finally, we would be implementing the framework for real-world deployment in the future and then perform extensive user studies with real users.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

## References

[1] B. Guo, D. Zhang, Z. Yu, Y. Liang, Z. Wang, and X. Zhou, "From the internet of things to embedded intelligence," *World Wide Web*, vol. 16, no. 4, pp. 399–420, 2013.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2013.

[3] P. Guillemin and P. Friess, *Internet of Things Strategic Research Roadmap*, Tech. Rep., The Cluster of European Research Projects, European Commission, Brussels, Belgium, August 2017, http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf.

[4] D. Hussein, S. Park, and N. Crespi, "A cognitive context-aware approach for adaptive services provisioning in social internet of things," in *Proceedings of the IEEE International Conference on Consumer Electronics*, Taiwan, June 2015.

[5] T. W. Um, G. M. Lee, and J. K. Choi, "Strengthening trust in the future social-cyber-physical infrastructure: an ITU-T perspective," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36–42, 2016.

[6] Q. Liu, H. Ma, E. Chen, and H. Xiong, "A survey of context-aware mobile recommendations," *International Journal of Information Technology and Decision Making*, vol. 12, no. 1, pp. 139–172, 2013.

[7] B. M. Sarwar, G. Karypis, J. Konstan et al., "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International World Wide Web Conference*, pp. 285–295, Hong Kong, May 2001.

[8] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Systems with Applications*, vol. 89, pp. 404–412, 2017.

[9] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, vol. 4321, pp. 325–341, Springer-Verlag, Berlin, Heidelberg, Germany, 2007.

[10] R. Burke, "Hybrid Web recommender systems," in *The Adaptive Web: Methods and Strategies of Web Personalization, LNCS*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., vol. 4321, pp. 377–408, Springer, Heidelberg, Germany, 2007.

[11] A. M. Otebolaku and M. T. Andrade, "Context-aware personalization using neighborhood based context similarity," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1595–1618, 2017.

[12] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[13] F. Ricci, L. Roback, and B. Shapira, "Introduction to recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Springer Science+Business Media LLC, New York, NY, USA, 2011.

[14] I. Mashal, T. Y. Chung, and O. Alsaryrah, "Toward service recommendation in internet of things," in *Proceedings of the Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference*, pp. 328–331, Sapporo, Japan, July 2015.

[15] M. A. Abbasi, J. Tang, and H. Liu, "Trust-aware recommender systems," in *Machine Learning Book on Computational Trust*, Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2014.

[16] U. Jayasinghe, A. M. Otebolaku, T. Um, and G. M. Lee, "Data centric trust evaluation and prediction framework for IoT," in *Proceedings of the IEEE/ITU Kaleidoscope: Challenges for a Data-Driven Society*, Nanjing, China, September 2017.

[17] H.-S. Choi, N.-S. Kim, and W.-S. Rhee, "Trusted cross-domain metadata model for context aware services," in *Proceedings of the Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 338–343, Milan, Italy, July 2017.

[18] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014.

[19] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[20] A. M. Otebolaku and M. T. Andrade, "Context-aware media recommendations for smart devices," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 1, pp. 13–36, 2015.

[21] G. Adomavicius, A. Tuzhilin, R. Sankaranarayanan, and S. Sen, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, no. 1, pp. 103–145, 2005.

[22] D. Margaris and C. Vassilakis, "Exploiting Internet of Things information to enhance venues' recommendation accuracy," *Jounal of Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 393–409, 2017.

[23] M. Tu, Y. K. Chang, and Y.-T. Chen, "A context-aware recommender system framework for IoT based interactive digital signage in urban space," in *Proceedings of the Second International Conference on IoT in Urban Space (Urb-IoT'16)*, Tokyo, Japan, May 2016.

[24] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[25] A. Chen, "Context-aware collaborative filtering system: predicting the user's preference in the ubiquitous environment," in *Proceedings of the International Workshop on Location-and Context-Awareness*, Oberpfaffenhofen, Germany, May 2005.

[26] A. Badii, M. Crouch, and C. Lallah, "A context-awareness framework for intelligent networked embedded systems," in *Proceedings of the Advances in Human-Oriented and Personalized Mechanisms Technologies and Services (CENTRIC) 2010 Third International Conference*, pp. 105–110, Nice, France, August 2010.

[27] Y. Saleem, N. Crespi, M. H. Rehmani, R. Copeland, D. Hussein, and E. Bertin, "Exploitation of social IoT for recommendation services," in *Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, USA, pp. 359–364, December 2016.

[28] C. Doukas and F. Antonelli, "COMPOSE: building smart & context-aware mobile applications utilizing IoT technologies," in *Proceedings of the 5th IEEE Global Information Infrastructure & Networking Services*, Trento, Italy, 2013.

[29] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[30] Wise-IoT, 2017, http://wise-iot.eu/en/home/.

[31] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall, and M. Zhao, "Standards-based worldwide semantic interoperability for IoT," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 40–46, 2016.

[32] E. Mingozzi, G. Tanganelli, C. Vallati, B. Martnez, I. Mendia, and M. Gonzalez-Rodrguez, "Semantic-based context modeling for quality of service support in IoT platforms," in *Proceedings of the 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, Coimbra, Portugal, June 2016.

[33] X. Amatrian, A. Jaimes, N. Oliver, and J. M. Puriol, "Data mining methods for recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., pp. 39–71, Springer, Berlin, Germany, 2011.

[34] A. M. Otebolaku and G. M. Lee, "Towards context classification and reasoning in IoT," in *Proceedings of the 14th International Conference on Telecommunications (ConTEL)*, pp. 147–154, Zagreb, Croatia, June 2017.

[35] A. M. Otebolaku and M. T. Andrade, "Context representation for context aware mobile multimedia recommendation," in *Proceedings of the 15th IASTED International Conference on Internet and Multimedia Systems and Applications*, Washington, DC, USA, May 2011.

[36] A. M. Otebolaku and M. T. Andrade, "User context recognition using smartphone sensors and classification models," *Journal of Network and Computer Applications*, vol. 66, pp. 33–51, 2016.

[37] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[38] C.-H. Liu, K. L. Chang, J. Y. Jason, and S. C. Hung, "Ontology-based context representation and reasoning using OWL and SWRL," in *Proceedings of the 8th Annual Communication Networks and Services Research Conference*, Montreal, QC, Canada, May 2010.

[39] D. Riboni and C. Bettini, "OWL 2 modeling and reasoning with complex human activities," *Pervasive Mobile Computing*, vol. 7, no. 3, pp. 379–395, 2011.

[40] A. M. Otebolaku, "Context-aware personalization for mobile multimedia," Ph.D. thesis, Faculty of Engineering, University of Porto, Porto, Portugal, 2015, https://repositorio-aberto.up.pt/bitstream/10216/78954/2/35037.pdf.

[41] Pellet, 2017, http://github.com/stardog-union/pellet.

[42] M. Bjelica, "Unobtrusive relevance feedback for personalized TV program guides," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 658–663, 2011.

[43] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Proceedings of the Second IEEE Annual Conference Pervasive Computing and Communications Workshops*, pp. 18–22, Orlando, FL, USA, March 2004.

[44] FIWARE NGSI, 2017, https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Monitoring_Open_RESTful_API_Specification.