A Human-Centered Decentralized Architecture and Recommendation Engine in SIoT

Daniel Defiebre[1], Dimitris Sacharidis[2*] and Panagiotis Germanakos[3]

[1]NextX, Kurfürsten-Anlage 52, Heidelberg, 69115, Germany.
[2*]Université Libre de Bruxelles, Campus du Solbosch – CP 165/15, Avenue F. D. Roosevelt 50, Brussels, 1050, Belgium.
[3]UX S/4HANA, Product Engineering, Intelligent Enterprise Group, SAP SE, P.O. Box 69190, Walldorf, 69190, Germany.

*Corresponding author(s). E-mail(s): dimitris.sacharidis@ulb.be;
Contributing authors: daniel.defiebre@next21.de;
panagiotis.germanakos@sap.com;

Abstract

The Internet of Things (IoT) enables smart objects to connect and share information, thus unlocking the potential for end users to receive more and better information and services. In the Social IoT (SIoT), objects adopt a social behavior, where they establish social connections to other objects and can operate autonomously in order to accomplish a given task. In this work, we present an SIoT architecture, called DANOS, based on three principles, dynamicity, decentralization, and anthropomorphism. Specifically, in DANOS (a) smart objects dynamically adapt their social neighborhood depending on the task, (b) information is decentralized and kept private, while efficient discovery mechanisms are prescribed, and (c) smart objects adopt a human-centered behavior determined by the personality traits of their users. We consider a general class of tasks that can be formulated as recommendations, and demonstrate how DANOS orchestrates the objects' social behavior. An extensive experimental evaluation validates our design choices, showing that three principles together result in improving the effectiveness of recommendations. The key lesson learned from our work is that SIoT architectures can benefit from the adoption of aspects of dynamicity, decentralization, and anthropomorphism.

Keywords: SIoT, Decentralized Architecture, Human-centered Models, Personality Traits, Smart Objects, Recommender Systems

1 Introduction

The Internet of Things (IoT) entails the interconnection of heterogeneous smart devices (or things) over a network infrastructure that facilitates their anytime and anywhere interaction and exchange of data [1, 2]. Current market analyses and studies predict that the IoT market size is expected to grow in an intense pace from USD 800 million in 2018 to USD 2800 million by 2023, at a compound annual growth rate of 28.4% during the forecast period [3]. Major growth factors may empower the creation of more stable and effective network infrastructures, where the interacting things will be able to maximize their outcome with minimal utilization of resources. The growth of the connected things should adhere to unique and defined network qualities like navigability, scalability, trustworthiness, and information availability, accessibility and shareability [4, 5].

IoT enables smart objects to be connected to each other and be discoverable over the confines of a single home or business environment. Just as humans benefit from establishing social connections, it is envisioned that the services enabled by IoT can similarly benefit if the smart objects adopt a social behavior [6]. This vision leads to the Social Internet of Things (SIoT) [4, 5], where each IoT object has a social agent that "socializes" with other objects' agents in an SIoT server (akin to a Facebook/LinkedIn service for objects), and forms friendship connections to selected other objects. The advantages that SIoT brings over IoT are that (a) objects and their services can be more efficiently discovered, and (b) trustworthiness can be established by enabling object owners to control the social relationships and flow of personal/sensitive information [7–9]. SIoT thus promises the proactive discovery (*what*) and delivery of information and resources *when* and *how* it is needed the most, while enabling a safe environment of interaction.

An example class of IoT services that can be greatly enhanced by the social aspect of SIoT is *recommender systems*. These systems help people make choices when the alternatives are too numerous or unknown [10], and have been extensively and successfully applied in various domains like e-commerce, entertainment, news [11–14]. Recommenders are known to be *data-hungry*; state-of-the-art methods are based on collaborative filtering principles and learn from millions of interactions from thousands of users. In an IoT environment, a recommender system, running on the application layer of a smart object [15], leverages the IoT network to collect *contextual* information that can be used to improve the quality of its recommendations. Apart from such contextual data, an SIoT recommender has access to an additional valuable information source: the interaction data from its social connections [16]. To further improve recommendation quality, an SIoT object can establish new connections (via the "socializing" mechanism of SIoT) with objects that contain information pertinent to the recommendation task.

In this work, we make two main contributions. First, we present a concrete SIoT system called DANOS, short for Dynamic and Anthropomorphic Network of Objects System, that follows the design principles and satisfies the requirements put forward by the SIoT vision [6]. Second, we present a recommender system built on top of DANOS that enables decentralized recommendations in IoT. We briefly overview the two contributions, relating them to current approaches and presenting their distinguishing aspects.

DANOS. While SIoT principles have been laid out for some time now [4, 17], there have been relatively few concrete instantiations of SIoT. DANOS is an SIoT framework that goes beyond state-of-the-art in three ways. First, the objects in DANOS establish and adapt their social connections *dynamically* depending on the task at hand and the characteristics of their social neighborhood. Specifically, objects maintain task-dependent measures of *neighborhood quality* that guide them in their decision to drop and/or look for new connections. Second, DANOS takes a *decentralized* approach locally, for information flow, and globally, for resource discovery. Locally, objects only exchange information within their friendship network, which dynamically adapts over time to improve task effectiveness. Globally, objects can efficiently identify relevant resources for a task thanks to DANOS' two-level organization into areas and cells, and the fact that cells *specialize* over time as they incorporate task effectiveness feedback. Third, the objects in DANOS are *anthropomorphic* in that they possess human-like traits. Given that objects in SIoT already have social behavior and act on behalf of humans, we posit that an object owner would be more satisfied by the services made available by an object, if the object exhibits a social behavior that is anticipated and resembles that of its owner. For example, an object of an extravert person would choose to socialize more with other objects, and hence be able to provide with diverse information and alternatives — an outcome that an extravert would be more likely to appreciate. We expect that the consideration of *individual differences* in the design and development of SIoT architectures, e.g., via the Five Factor Model (known also as the Big Five Inventory, BFI) [18], would add significant value on the quality and performance of a generated social network.

Decentralized Recommendations. We design a recommender system that fully exploits DANOS. Its distinguishing aspect is *decentralization*, which improves recommendation *effectiveness* while ensuring the *privacy* of object owners. Existing IoT recommenders, e.g., [15, 19–22], either (a) learn only from interaction/feedback data originating from the object's owner, limiting thus their effectiveness, or (b) operate on a centralized database where each object openly publishes their owner's data, sacrificing thus their privacy. In contrast, DANOS enables the objects to establish purposeful social connections in a decentralized, private, and controlled manner, so as to exchange only data that are useful for the recommender and authorized by its owner. Outside the IoT context, our recommender is related to *social-aware* (e.g., [23–25]) and *decentralized* (e.g., [26–28]) recommenders. Besides the apparent fact that none of them is designed for IoT, the former line of research differs from ours in that they only consider a static social network and do not address neighborhood formation, while the latter line differs in that they either assume adhoc (i.e., non-malleable) networks or require objects to store other objects' data to implement a distributed hash table.

Running Example. To make the aforementioned concepts and ideas more concrete, we introduce an IoT scenario that we follow throughout the paper. Consider a smart home setting, where a thermostat is IoT enabled, i.e., connects to other smart objects in the home and to the internet. The thermostat is a smart physical object [29] and has knowledge of the current context, including date and time, the weather conditions (via a weather sensor), and the situation in the house, e.g., if occupied and by how many (via presence sensors, actuators, smart lights, smart lock, etc.). The thermostat makes temperature recommendations to its owner with the goal of providing a pleasant environment while minimizing energy consumption. These recommendations are generated by analyzing the feedback provided by the owner (i.e., manual temperature adjustments) and the contextual information. IoT is the enabling technology that drives this scenario.

Now consider the SIoT setting, where the smart thermostat acts as a social agent on behalf of its owner to achieve its goal (pleasant environment with minimum energy consumption). By employing the SIoT functionality exposed by DANOS, the thermostat is able to identify and connect with other thermostats with similar profiles (e.g., in terms of number and size of rooms, household composition, location, home energy profiles, presence patterns), and also exchange feedback information (subject to owners' consent). These additional data sources enable the thermostat to draw novel inferences regarding temperature levels and energy consumption, which lead to temperature recommendations that further reduce the energy consumption.

This example is revisited in Section 4.1, where we describe in detail how DANOS processes such a scenario.

Design Principles. Our work is guided by our ambition to implement and improve upon the original SIoT vision [6]. We design DANOS based on three key *principles*:

- *Dynamicity.* Given a specific goal (e.g., a recommendation task), objects in DANOS form social neighborhoods that continuously adapt over time so as to increase their utility. This is achieved by a mechanism that allows objects to evaluate their neighborhood utility based on their owner's feedback, and to decide to drop or establish new connections.

- *Decentralization.* Information possessed by objects is not publicly shared in the IoT network. Instead, objects are only allowed to exchange information with those objects that are socially connected to. Because object owners have complete control over which social connections their objects are allowed to make, the owners essentially control the flow of data; e.g., an owner may allow connections only across objects they own. Decentralization necessitates an efficient discovery mechanism to identify objects that contain potentially useful resources. Specifically, DANOS introduces a novel mechanism, based on *cells*, essentially meeting rooms for objects, which become specialized over time by incorporating the feedback from useful object-object connections.
- *Anthropomorphism.* Objects in DANOS inherit the human factors of their owners, specifically their personality traits, which are manifested as distinct human-centered behavior (HCB). This HCB dictates how objects socialize. For example, an object with a risk-taking behavior is more likely to connect with other objects that are not as similar.

DANOS aims to be *effective* in processing IoT tasks thanks to the aforementioned design principles. Specifically, the discovery mechanism of DANOS paired with the dynamicity of the objects' social connections and their anthropomorphic behavior counteract the constraints a decentralized platform introduces.

Current SIoT Challenges. We note that the aforementioned design principles essentially address the current *challenges* in SIoT as distinguished in [30], and in fact, they match the proposed *solutions*:

- *Heterogeneity of SIoT networks.* DANOS addresses this challenge via anthropomorphism and decentralization. It acknowledges the non-homogeneity of objects by endowing them with human-like traits. Moreover, it introduces a task-oriented two-level virtual space, termed *cyberspace*, for objects to explore and self-organize. The top level consists of *areas* that correspond to different tasks (e.g., the area for temperature recommendations), while the bottom level consists of the *cells*, the object meeting rooms. This approach matches the middleware or interface ideas presented in [30].
- *Mobility and Dynamicity.* DANOS addresses this via dynamicity. It allows objects to adapt their social connections and visit cells to acquire better connections. [30] proposes the introduction of objects communities, which is exactly what the specialization of cells enables.
- *Tracking Objects.* DANOS addresses this via decentralization. Specifically, it tracks the movement of objects in the virtual space, and their successful interactions in cells. Based on this information, DANOS is able to prescribe an efficient *travel schedule* to objects in their quest to locate useful resources. This is in line with setting rules and object movement patterns proposed in [30].
- *Security, Trust, and Privacy.* DANOS address this via decentralization. By design, it employs an access control mechanism that gives total control of

objects to their owners, and only allows the flow of information among socially connected objects. This aligns with the access control system and safe data sharing model in communities of trust proposals in [30].

- *Resource-Constrained Devices.* DANOS address this via dynamicity. Specifically, it allows objects to form small social *neighborhoods* whose quality is optimized over time via an *online learning* approach. This resonates with the call for effective resource management systems in [30].
- *Efficient Service Search and Discovery.* DANOS address this via decentralization. More precisely, it exposes a sophisticated two-level virtual space and travel schedulers for objects to efficiently discover other objects. Interestingly, among the prescribed solutions in [30], is a call for better navigability and the creation of object communities, which are at the heart of DANOS.

Research Questions. The basic principles and ideas of DANOS were developed over a line of work [31–34]. In this paper, we present all aspects of DANOS in detail, and moreover we seek to validate the design choices by considering a concrete IoT task, making recommendations. Therefore, we pose the following research questions.

RQ1. How good are the neighborhoods formed in DANOS? (*Dynamicity*)
RQ2. How efficient is the discovery mechanism of DANOS? (*Decentralization*)
RQ3. What is the effect of the human-centered behavior in DANOS? (*Anthropomorphism*)
RQ4. How effective is DANOS in processing IoT recommendations? (*Effectiveness*)

Note that we do not explicitly investigate the privacy aspect because DANOS is by design private: IoT data in DANOS is only transferred between socially connected objects under the authorization of their owners. Instead, we investigate the implications of this design choice in RQ3, when we compare task efficiency in DANOS (a decentralized approach) and in a no-privacy setting where all objects' data is openly published (a centralized approach).

Later, in Section 5, we make these questions more concrete by considering the specific characteristics of DANOS and the task at hand, recommendations.

Outline. The remainder of this paper is organized as follows: In Section 2, we present previous related research attempts related to SIoT networks and recommendations in IoT. In Section 3, we present the components of DANOS. Then in Section 4, we focus on a specific task, recommendations, and show how it can be executed in DANOS. Section 5 presents the experimental evaluation of our architecture and recommender, while Section 6 concludes this paper.

2 Related Work

Our work makes two main contributions: (a) it introduces DANOS, a novel SIoT system driven by the current challenges in IoT and SIoT, and (b) it presents a recommender that exploits DANOS to improve upon existing IoT recommenders. Therefore, this section classifies related work along these two lines of research. It first presents work on IoT/SIoT networks and on how human aspects relate to SIoT. It then presents recommendation approaches in general, in a social context, and in decentralized environments, before discussing IoT-specific recommender systems.

2.1 Social Internet of Things

IoT and SIoT Networks. The evolution, utilization and growing ubiquity of heterogeneous technologies (such as, sensors, bar codes, Near Field Communication (NFC) tags, RFIDs, 3G/4G, 5G, NB-IoT, LTE-M network connectivity, etc.) along with the growing demand of services automation during the last years, generated a number of opportunities for new concepts and infrastructures to arise. One of them is the IoT which refers to uniquely identifiable objects and their virtual representations in an Internet-like structure [35]. The IoT might refer to a connected world based on the advancement of a number of intelligent devices and services that bring smartness in an ecosystem, enhancing communication, increasing speed, social inclusion, etc.

Inevitably, aside from the undeniable benefits that this new technological reality bears, it may also create an uncertain situation around the users that are likely to experience difficulties or discomfort to execute daily tasks, operations, or to control and maintain an end-to-end understanding of processes that are triggered from specific requests. Such a reality, at a far extreme, is liable to affect their (subjective) perception and might create intense and hard to deal interaction scenarios since the features and functionality of this 'social network of things" might generate unfamiliar flows of information from one smart object to another difficult to comprehend. In situations, the communication entails another element of interaction, the social agents (as agents we mean entities that might represent i.e. humans, machines, robots, intelligent proxies and algorithms), which are responsible to communicate directly with the objects, update the profiles, friendships, and to discover and request services for the social network [4]. When an agent behaves like a human and acts on his behalf (i.e., intelligent agent) in various circumstances, it means that the agent inherits several characteristics and features of its owner, referring to the humanization of the agent [36]. Essentially, agents are processes that aim at performing tasks for their users, usually with autonomy, playing the role of personal assistants [37]. Some of their main characteristics could be distinguished according to their abilities (such as intelligence, autonomy, social capacity – inter-agent communication) and according to the tasks they execute (may be classified into information filtering agents, information retrieval agents, recommendation agents, agents for electronic market, and agents for network management [38]).

In recent years, the research in the area of SIoT – as an emerging paradigm of IoT that enables users and smart things to interact and collaborate in a framework that imitates relationship models, operations and features of a HSN [39], has been gaining increasing popularity due to the alternative and flexible opportunities that offers to users with respect to potential overwhelming scenarios. It draws attention on the creation of innovative models, architectures and applications that will be able to handle the huge amount of IoT objects and will deliver more effectively and efficiently services to their users, by considering the "social" interaction of heterogeneous smart devices over integrated networks, fulfilling their requirements and intents. In principle, the literature approaches the SIoT area from different angles. It emphasizes on its definition, the relationship it has with the IoT, and its expected impact on specific ecosystems. It presents alternative architectures and functionalities regarding information and services crawling, and delivery; it details the characteristics of interactions between smart objects; it discusses principles that may influence the user to object relationship, and finally, protocols and APIs that could ensure a smooth communication between humans, things and services. More specifically, a comprehensive work from Atzori et al., (2012) [17] detail the policies for the management of social relationships between objects and propose an architecture with functionalities liable to integrate things into a social network; Cheng et al., (2014) [40] propose a framework that facilitates the harmonious social networking among human, things and services; Ding et al. (2010) [41] describe information (via Internet), people (via HSN) and objects (via object networks) as macro elements of the human society and discuss how they can co-exist under a common framework of interaction, proposing a platform that can cluster the three together and provide the means for further observation and experimentation of the produced behaviours; Kim et al. (2015) [42] propose a system, called Socialite, that enables new IoT applications based on emerging types of social relationships and semantic models for SIoT that include device types and their capabilities, users and their relationships; Cena et al., (2019) [29] propose a framework that drives the understanding of how a smart physical object can be designed in relation to specific requirements, abilities and dimensions that define its nature, level of smartness (in terms of cognitive and interaction – social – abilities) and functioning. The current framework can be used as guidelines for building smart objects or to facilitate their classification for comparison; Kranz et al. (2010) [43] emphasize on the combination of social and technical networks to collaboratively provide services to both humans and systems; while Guinard et al., (2010) [44] propose a platform that enable people to share their services and devices so that others, people or things, can use them. In addition, authors in [45] introduce the small world concept in SIoT by integrating properties of the former with the latter paradigm. Their proposal suggests that the use of smart social agents may ensure the finding of appropriate friends and services required by the user without human intervention. Kasnesis et al., (2016) [46] propose also an architecture and simulator (ASSIST) for supporting SIoT with the use of agents that enable device-to-device and human-to-device social communication for discovering services that satisfy the users needs.

SIoT and Human Factors. To the best of our knowledge, there is no SIoT research or architectures that employ individual differences — specific human characteristics or models from the area of e.g., Cognitive or Social Psychology, and which have been proved successful in the Human Social Networks (HSN) — recognizing always the situation-specific perspectives and interpretations. Most of the works that approach the topic of a Humanized IoT (HIoT as an umbrella term that includes SIoT and Internet of People (IoP) [47, 48] approach the topic from a more theoretical stance, defining conceptual models, profiles and frameworks rather than explicitly exploiting the potential that given human factors or traits like e.g., motivation, creativity, personality, stress, might bring into the interactions of a system as a proactive process step.

Only a few works bring the human more prominently into the center of attention for building solutions and functionality that are driven from the modeling and impact of specific individual traits. Such research, either discusses the topic from a more theoretical perspective, capturing the breadth and depth of the concept and the association of its elements, or attempts to establish a practical framework of application. Indicatively, Pintus et al. (2015)[47] explain how the HIoT can bring a more human perspective to the technology by including Fiske's [49] four common forms of sociality that people use in their relations. Koreshoff et al. (2013) [48] investigate a more human-centered perspective of IoT through the lens of HCI, attempting to convey a less technical perspective of the elements that influence the interaction of people with the various technologies. More recently, Ursino and Virgili (2020) [36], introduce the Multiple IoT (MIoT) paradigm and architecture as well as the notion of "humanization" of things in complex scenarios where things are organized in several IoTs cooperating with each other. It discusses the composition of user and things profiles and how they influence the interaction process in MIoT while maintaining historical and neighborhood data, and they investigate the concept of reliability of a thing in IoT. Jung et al. (2018) [50], propose a model (Social Strength Prediction Model) that builds upon the social relationships of smart objects in SIoT for constructing social networks. Main contribution of the model is that it computes the entropy-based and distance-based social strength of objects, capturing different properties like diversity and spatiotemporal features. Lastly, Roopa et al. (2018) [51], use physical location proximity and social context of users in social communities to facilitate object search in SIoT.

Nevertheless, by definition users have unique traits, abilities, experiences, etc., that directly affect their expectations, actions and decisions. Thereof, it is vital to regard individual differences in the process of information search, retrieval and delivery in an SIoT ecosystem in order to increase users' cognizance, usability and reliability. The term individual differences is very broad, since it could include from genetics to personality. It was proposed by Stern (1900) [52], in order to summarize the research on mental differences, in coordination to a notion of "general psychology". During the years, the emergence and proliferation of the individual differences research in relation to information technology is mostly linked to the study of intelligence (recognizing cognitive factors and abilities) [53] and emotions [54] in information processing.

Personalizing and adapting users' navigation process, information presentation and recommendations based on specific human factors may increase performance, accuracy and satisfaction even with respect to highly complex and intense environments [55], as is SIoT settings. This could be achieved by defining more inclusive user-agent (or object) models, extraction and interpretation methods for building more realistic SIoT simulations. Accordingly, considering intrinsic human factors such as visual, cognitive and/or emotional characteristics for the definition of interaction processes and solutions, apart from the traditional profile elements of the user (i.e. age, experience, profession, tasks, interests, time, location), or the object/ channel characteristics (i.e. displays, connectivity, processing power, interface and data entry).

The relationship between smart objects/ agents, individual differences and SIoT ecosystems, to our knowledge, is largely unexplored. There is a lack of research that exploits the imminent utilization of human factors, apart from theoretical references and constructs, directly in the formulation of models and rules that would drive the interactions of agent-based SIoT (or HIoT) solutions and platforms. In this respect, distinctive values that differentiate the current work from related, is that it employs specific human factors (for the purpose of this paper we use the personality traits), in the core of the proposed DANOS SIoT architecture, guiding the creation of objects' profiles, their human-centered behavior during navigation and matching while running over a decentralized network, as well as the provision of best-fit recommendations to their owners (users), considering their direct feedback to the evaluation and optimization of the algorithms and the results.

The theory of personality types could be regarded as a well-known, comprehensive theory that refers to individual differences in preferences, behaving, thinking and feeling [56]. More broadly, personality traits (which belong to one of the nine categories of personality, namely Psychoanalytic category) are rather predictable and remain stable [57] over time, and emphasize on the ways people differ psychologically from one another and how these differences might be conceptualized or measured [58]. They can be acquired explicitly with the use of psychometric tests, usually in the form of questionnaires, or implicitly using digital imprints (e.g., from social data [59]) or other real-time settings (like video games [60]) and by applying regression or classification techniques to assign meaning. Such meaning can successfully facilitate the recommendation and personalization of content and services, as it is evident from their extensive utilization through various scenarios and applications in HSN. In this respect, It has been widely appreciated and utilized by researchers to explain behaviors and patterns of users when interacting with the content and peers in HSNs, with really encouraging results and proved impact in the respective domains, such as Facebook [59, 61, 62], Instagram [63] and Twitter [64] as well as on the internet [61, 65] and human social relations [66].

Discussion. Our proposed system, detailed in Section 3, builds upon the SIoT concepts [4] that all aforementioned SIoT architectures follow, but differs in three key aspects: *dynamicity*, *decentralization*, *anthropomorphism.* (a) Prior work in SIoT architectures acknowledges a passive notion of dynamicity, where social connections among objects can be ephemeral, e.g., due to co-location. In contrast, key in DANOS is an active notion of dynamicity, where each object can autonomously decide to terminate or establish friendship connections so as to better accomplish its task; the decision depends on task-specific notions of quality for the object's friends. (b) Prior work in SIoT follows one of two opposite paradigms when it comes to information sharing. One paradigm is to publicly share information on the internet so that all objects can access it. The other paradigm is to only share information between socially connected objects. DANOS follows the latter paradigm, but extends it with a decentralized discovery mechanism that enables objects to identify and connect with those other objects that contain relevant information. (c) Current work in SIoT do not consider specific individual differences in the design or execution of their models or technical approaches. In contrast, we adopt a human-centered standpoint to facilitate object-to-object interactions and develop effective relationships among objects increasing the effectiveness to the task at hand. The effects of these three distinguishing aspects to state-of-the-art are experimentally investigated in Section 5 for a recommendation task, where it is shown that their adoption leads to a better utilization of information and network resources, to the optimization over time of the quality of information that is exchanged among objects, and ultimately to a higher task effectiveness.

2.2 Recommender Systems in IoT

Recommender Systems. The benefit of recommender systems has been widely recognized by researchers in the last decades [10]. They refer primarily to computational routines and systems that collect recommendations from a number of people around topics or items of interest and then after applying various aggregations they deliver new recommendations to others, targeted people, helping them to make decisions. These recommendations (usually useful lists of items with ratings of similar users or items used in the past) may refer to individuals or group of people (group recommenders) based on specific characteristics and preferences. Traditionally, there are 3 major types of recommender systems: (a) Collaborative filtering (CF), which may be considered as the most widely exploited domain of recommenders, with mature technologies that aggregate ratings or recommendation of items by recognizing common neighbor users' behaviors with respect to their ratings and produce new recommendations based on users' similarities; (b) Content-based filtering uses existing items and ratings by users and compares them to targeted users' interests (found usually in user profiles) and items that they have already consumed; (c) Knowledge-based recommenders refer to a family of recommenders that make recommendations of specific items based on related inferences of users' preferences and needs. Other recommenders include demograhic- or utility-based recommenders as well as hybrid recommender systems that might combine in situation-specific scenarios the strengths of two or more recommender techniques, e.g., computing a weighted result once they have received the outputs of a collaborative filtering and content based filtering recommender system [11–14].

Social-Aware Recommender Systems. In social- and trust-based recommenders, users are connected to each other with friendship and trust-based relationships, respectively. The distinction is often subtle; in the latter case, the users explicitly indicate to the system that they trust the preferences of other users. In any case, the system exploits the social connections to deliver more effective recommendations. This is motivated by the mechanisms of *homophily* and *social influence* observed in social networks [67], which suggest that our preferences and tastes tend to become similar to those of whom we interact with in our everyday life [68–70].

Typically, social-aware recommenders extend collaborative filtering (CF) techniques, using information from two sources, user-item interactions (as typical in CF recommenders), and the adjacency matrix of the social network. As in CF methods, a distinction can be drawn between memory-, or neighborhood-based, and model-based social-aware recommenders.

Early work on social-based recommenders was based on memory-based CF, and focused in social connections conveyed trust between users of the system. In [71, 72], the authors propose a memory-based CF technique to integrate trust into recommendations, which is called Trust-aware Recommender System (TaRS). Starting with a graph induced by explicit trust statements, one can define *local* and *global* metrics to quantify the trust between any two users. The former compute a subjective measure of trust, while the latter an objective measure of global reputation. In [72] the same authors present detailed evaluation results of their technique, which implements a simple local trust metric, called MoleTrust. The proposed algorithm predicts the rating based on a user-based CF technique, where instead of the user similarity, the user trust is used to determine the neighborhood and weigh the ratings. In all experiments, this technique resulted in higher accuracy (in terms of maximum absolute error) and coverage (in terms of number of predictable ratings) than standard user-based CF. Also, they find that hybrid techniques based on trust and similarity, and global trust metrics (such as PageRank) performed worse than pure local trust-based ones.

More recently, social-aware recommenders are based on matrix factorization (MF), model-based techniques [23, 24, 73, 74]. The most prevalent technique in this line of research is *social regularization*. Briefly, the idea is that the latent representations, extracted by the factorization process, of socially connected users should be similar [24, 74]. The degree of regularization between two users could be controlled by the degree of their rating/feedback similarity. Several variations on the basic idea of social regularization have been proposed since then [75–79]. The current state of the art method extends the local low-rank matrix approximation ensemble method [25] in two ways: (1) the users and items comprising a local model are determined by the social network structure, instead of user-user and item-item rating similarities, and (2) pairwise social regularization is employed.

Decentralized Recommenders. Decentralized recommenders use information that is distributed over a network of several nodes (peers). Each peer has only a partial, local, view of the information in the network, and is connected to other peers, from whom it may retrieve additional information. We classify decentralized recommenders according to the dynamicity of the network established.

First, we consider the case when the network is fixed, i.e., does not change or evolve over time. The works in this class essentially employ ideas from the literature of social-aware recommendations. For instance, [80] computes trust weights among peers, and then uses these weights in place of similarities in a user-based collaborative filtering (CF) technique, similar to [72]. Specifically, the type of information exchanged is a user profile (a history of ratings/feedbacks about some items), and a peer may request information from other peers up to some number of hops away. All collected profiles are weighted by the computed trust values and aggregated to predicted ratings/interest in unknown items.

In some other works, [26, 81] peers form connections via epidemic, or gossip-based communications with the goal of connecting to similar-minded peers. For this purpose, there is a similarity function involved, which typically computes rating/feedback similarity. After the network is formed, however, it is treated as fixed, i.e., the network will not change over time as recommendations are made and feedbacks are received. [81] employs a user-based CF techniques. The feedback from peers up to two hops away are collected (friends and friends-of-friends). Then a random-walk approach computes an adjusted similarity value between two peers, which is used to weigh the ratings. [26] employs a simpler recommendation engine, where peers simply send a list of recommended items, rather than their profiles.

Another approach is to connect peers with a distributed hash table (DHT) approach. In this case, the network is again fixed, but the DHT dictates how data is stored and connections are made. This means, that peers do not have control over their data, which raises privacy and security concerns. For instance, in [82], a peer uses a DHT to locate its most similar peers, retrieves their ratings, and then recommends using a plain user-based CF technique.

All aforementioned techniques are essentially a memory-based CF, where the information exchanged is the rating profiles. In contrast, in model-based CF techniques, like matrix factorization, the information exchanged is a local view of the model. However, for such an approach to work, each peer must store not only its own ratings, but also a part of the global rating matrix to be factorized [27]. Thus, privacy and security issues are also raised in this case.

There is another line of work, [28, 83, 84], where peers establish device-to-device (D2D), connections in an opportunistic manner, e.g., when they are in close proximity to each other. Therefore, the network continuously changes and the peers have no control over it. In this setting, all a peer can do is collect information, rating profiles, from the peers it has connected to it at some point in the past. Typically, user-based CF is used to make recommendations based on the collected profiles [28, 83]. [84] focuses on how to establish such D2D connections from a technical, networking standpoint, and is not concerned that much with recommendation techniques.

Recommendations in IoT. IoT recommenders have been proposed for various application domains, e.g., for recommending IoT apps or services [15, 19, 20], for personalized shopping [21], in smart homes [22], in technology fairs [85], in museum visits [86], in sports events [87], for assisted living [88]. Refer to [15, 89] for a detailed overview of recommendation techniques suitable for IoT. Specifically, [15] discusses collaborative, content-based, utility-based, sequence-based, constraint-based techniques that can be applied as is. Moreover, [15] also presents some novel adaptations of recommendations methods for IoT. In general, IoT is seen as the means to collect more data about the user, such as the context [90], so as to provide better domain-independent recommendations.

In IoT, it is often desirable to recommend services offered by other objects. [20] studies the case where a service is to be offered to a group of users. Thus, a group recommendation approach is proposed, which is in fact independent of the IoT setting. As another example of service recommendation, [19] employ random-walk techniques, like PageRank, over a tripartite graph defined by owner-object-service relationships.

There are some few works that specifically target the SIoT domain. [16] discusses ideas and challenges of developing a recommender in SIoT, but offers no concrete implementation. [91] presents a discovery mechanism, rather than a recommendation engine in SIoT. A fixed network of agents is assumed, where the agents exchange their data (items to discover) so that eventually nearby agents posses similar data. Exchanging data instead of adapting social connections poses security and privacy threats. Hence, in our work we establishing social connections between the objects of the network by calculating their similarities using a third computational entity, i.e., relationship manager, ensuring that only the ratings will be shared among the objects in a pseudonymized manner.

There is another line of work addressing neighborhood formation (sometimes called friend recommendation) in SIoT. [92] discusses a task independent approach, where feedback from previous transactions among objects is used to define a metric suggesting when a friendship is to be established or cancelled. This idea of object-to-object transaction feedback also appears in our recommendation engine: feedback from the user on an a specific item, affects the similarity between the user's object and other objects. Neighborhood formation in SIoT may also depend on trust relationships among objects, e.g., as defined in social-aware recommenders. [93] discusses how to compute and maintain trust using transitive relationship, and also how to infer trust for unseen tasks. In our system, a similar idea is conveyed by the object-object similarity metrics, which evolve over time based on users' feedbacks.

In conclusion, we note that ours is the only work that considers domain-independent recommendations in SIoT, proposes a concrete solution characterized by a novel neighborhood formation mechanism, and evaluates it in an SIoT simulator.

Discussion. In the context of IoT or SIoT, existing proposals for implementing recommender systems exploit either just local data or adopt a centralized approach, where all objects publicly share relevant information in the internet. Approaches of the latter type come in direct contrast to ours, where we advocate a decentralized flow of information controlled by object owners. The disadvantage of decentralization is that the recommender has potentially access to less data. To compensate for this, our SIoT recommender leverages the dynamicity and decentralization features of DANOS to efficiently identify the most relevant information. In Section 5.4, we demonstrate that our recommender can be as effective as the centralized approach.

Compared to other decentralized non-IoT recommenders, we find that they either assume a data organization structure (e.g., a distributed hash table), or they consider a fixed decentralization network. Methods of the former category are not compatible with the heterogeneity and loose coupling of IoT objects, as they require each object to accept storing other objects' data and be always available to process requests. Methods of the latter category, however are compatible with IoT, and operate on similar principles to social-aware recommenders where a static social network is assumed. In contrast, DANOS dynamically adapts the social connections based on feedback from the recommendation task so as to continuously improve effectiveness. In Section 5.4, we demonstrate that our recommender is more effective than a static approach.

A final note concerns the recommendation model employed in DANOS. Recall that each object receives relevant information from its social neighborhood thanks to the DANOS functionality. Based on the available information, the object then builds a recommendation model and processes requests. We emphasize that any recommendation model can be utilized, and the choice of a particular one is orthogonal and agnostic to DANOS. In our discussion in Section 4, we consider a memory-based collaborative filtering recommender; the presentation is easily generalizable to other recommendation models. The important aspect of our work is how to ensure that each object creates useful social connections and maintains them over time as recommendation feedback is collected. These functions are the responsibility of DANOS, and the recommendation task serves simply as an illustrator of DANOS functionality.

3 The DANOS Architecture

SIoT architectures extend the functionality of IoT networks by allowing objects to adopt a social behavior and act autonomously. Therefore, the distinguishing aspects of SIoT — compared to a regular IoT architecture — concern the management of social connections. Although it is hard to discern a common design in the literature, current research [4, 30, 42, 94] agrees that an SIoT architecture should: (a) define object profiles, (b) define object relationships, (c) manage relationships, and (d) enable owner control.

DANOS is an SIoT architecture designed upon three principles, *dynamicity*, *decentralization*, *anthropomorphism*. It improves existing SIoT architectures in each of the aforementioned SIoT aspects. Specifically, DANOS endows object with human factors (a), which enables anthropomorphic behavior and owner control (d). Moreover, DANOS defines dynamic and intent-oriented object relationships (b), and manages relationships in a decentralized manner (c).

In the following, for each SIoT aspect, we explain its function, and present the specific contributions DANOS makes. We reference the various components of DANOS, depicted in Figure 1 and summarized in Table 1.

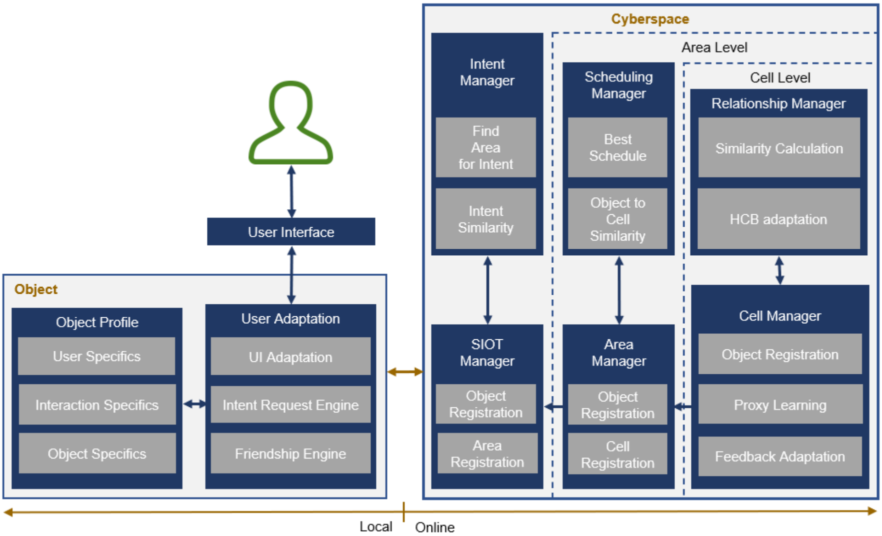


Fig. 1: The DANOS Architecture

3.1 Object Profile

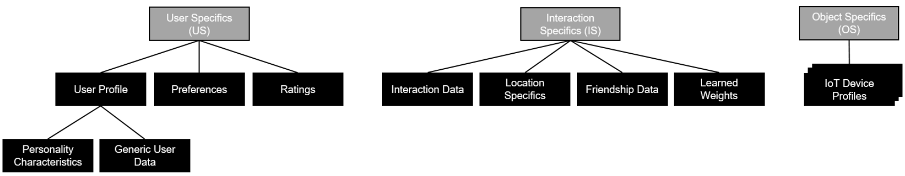


Fig. 2: Parts of the Object Profile

SIoT Function. The object profile is the static and dynamic information associated with objects, based on which object relationships are established [4]. As an example, for a smart thermostat, the object profile would include vendor information, sensor characteristics, configuration profiles, household composition, etc.

Implementation in DANOS. In DANOS, we refer to the standard information in SIoT object profiles as *object specifics*, and extend the profile to also include *user*- and *interaction specifics*, as shown in Figure 2.

User Specifics describe the user and consists of the User Profile, Preferences, and Feedback/Ratings. The User Profile consists of Personality characteristics and other inherent values as those dictated by the Theory of Individual Differences like cognitive, emotional, perceptual, and generic user data like age, gender. Those factors are inherited from the user through the initialization stage of the process and will be dynamically maintained (accommodating any changes e.g., on user's preferences and feedback).

From the number of the existing personality models we qualified Goldbergs's Big 5 [95]) personality construct which consists of five main factors describing the personality of an individual, outlined as follows:

- *Extroversion:* is characterized by excitability, sociability, talkativeness, assertiveness, and high amounts of emotional expressiveness. People who are high in extraversion like to start conversations, enjoy meeting with people and sharing news.
- *Agreeableness:* reflects much individuals' dimension that includes attributes such as trust, altruism, kindness and affection. People who are high in agreeableness tend to be more cooperative, enjoy helping and contributing to the happiness of other people.
- *Conscientiousness:* include high levels of thoughtfulness, with good impulse control and goal-directed behaviors. Highly conscientiousness tend to spend time preparing, be organized and pay attention to details.
- *Neuroticism:* is a trait characterized by sadness, moodiness, and emotional instability. Individuals who are high in this trait tend to experience mood swings, anxiety, get upset easily and worry about many different things.
- *Openness to Experience:* is a trait that is characterized by imagination and insight, and those high in this trait also tend to have a broad range of interests, to be more open to try new things, tackle new challenges and are welcoming thinking of new concepts.

Our decision to use the aforementioned personality model, apart from its wide acceptability in the research community, was based on the fact that we would expect to meaningfully interpret the characteristics and attitudes derived by the interrelations and the combination of the various personality factors, and accordingly to define behaviors and actions that we would be able to fuse into the agents assimilating a human-centered like interaction while executing purposeful tasks (see Table 2 for their adoption in the SIoT domain and definition of object behaviors).

The *Interaction Specifics* describe the data that is generated through the object's experience in the network during their lifetime. It consists of the following parts:

- Interaction Data: Stores generated data in the network, e.g. how many times an object had crawled for new friends to create the neighborhood, how many times it was active.
- Location Specifics: Stores cell-related data, e.g. how many cells the object had visited, in which cell it is currently registered, the current travel schedule.
- Friendship Data: Stores information about the local neighborhood, which is a list of other objects with which the object can directly communicate and exchange information (ratings/feedback).
- Learned Weights: Stores parameters that control the neighborhood formation and are learned over time.

The *Object Specifics* describe the object's specifications using various static attributes that only relate to specific devices (e.g., specifications of a smart TV, installed apps on a Smartphone) and without any interaction or user data.

3.2 Object Relationships

SIoT Function. Object relationships are at the heart of SIoT. There are different types of relationships that SIoT object might have: parental/family, between objects of the same manufacturer; owner, between objects owned/-operated by the same person/organization; co-work, between objects that collaborate to achieve a specific task; co-location, between objects operating in the same physical environment; and social ,between objects that sporadically or continuously come into contact with each other.

Implementation in DANOS. DANOS dictates co-work object relationships. One of the main contributions of DANOS is its dynamicity: it establishes and removes relationships based on the feedback to the task at hand, the intent. In our running example, the intent of the smart thermostat is to save energy, so it looks to become friends with those objects that would help it best achieve its goal. Based on user feedback, the thermostat may decide to drop some friendship connections and seek new co-work relationships via the relationship management process, described in the next section.

The DANOS components that enable dynamicity are the following. The *Intent Request Engine* takes the user's intent (e.g., set the temperature), checks if the local neighborhood (part of the object's interaction specifics) is of sufficient quality to handle the intent. It triggers the Friendship Engine to establish new friendships if quality check fails. The *Friendship Engine*, is responsible for (a) exchanging information among the local neighborhood, and (b) initiate the search for new co-work relationships, and establish new relationships. The *UI Adaptation* (a) interfaces with the owner to present the task result and adapt it based on the owner's personality characteristics, and (b) receives feedback, which becomes part of the user specifics part of the object profile.

3.3 Relationship Management

SIoT Function. The relationship management is the key component of SIoT that enables object to autonomously connect with each other. This is possible thanks to the information stored in the object's profiles. In our running example, the thermostat establishes co-location connections to all other objects (e.g., weather sensors, energy monitors, home security sensors) that reside in the same house and are connected via the same IoT gateway. Moreover, the smart thermostat may establish social connections with the owner's mobile devices, such as a smartphone, smartwatch, that are occasionally co-located.

Implementation in DANOS. The management of co-work relationships has not been concretely addressed in prior SIoT architectures. For relationship management, DANOS adopts a two-level *decentralized* approach, termed the *cyberspace*, i.e., where objects socialize. When triggered by the friendship engine, objects enter the cyberspace in order to discover other objects that may possess useful resources. In our running example, an object may decide to look for other objects with similar energy profiles, habitation patterns, household composition, etc.

The DANOS cyberspace is organized into two levels. *Areas* are intent-specific, e.g., there is an area for the intent of setting the temperature in homes. Each area is further organized in *cells*. Cells act as virtual meeting rooms for objects, in analogy to how people frequent locations that they prefer. Cells evolve as objects visit them, and over time they *specialize*, meaning that they tend to be visited by objects having similar profiles. In the following, we describe the various components in DANOS cyberspace, as depicted on the right-hand side of Figure 1.

The *SIOT Manager* is responsible for the registration of the objects and the areas. It represents the entry point for objects that they can use the functionality of the cyberspace. As areas can run on different machines, they have to register themselves on the SIOT manager. Through this registration, the Manager is able to forward request from objects based on an intent to the specific area. The *Intent Manager* provides a directory which maps the intent with the areas addresses. When an object wants to handle a specific intent in DANOS, it contacts the Intent Manager which calculates a Intent Similarity from the object's intent to the directory based on natural language processing techniques. If the similarity with known intents is low, the object can be notified to request from its user to redefine the intent. Or in the case of very low or no similarity, a new intent is recognized and the SIOT Manager assigns an area to handle this new intent. Intents can also get obsolete if no objects requests for them. In this case, the SIOT Manager assigns the area to another intent.

The *Area Manager* has an address (which the SIOT manager forwards to the objects), so that objects can find it and register. The area manager handles a directory of cells with their addresses, along with their proxies (see Section 4.2). The Schedule Manager computes the *best travel schedules* for objects, i.e., a list of cells an object should visit in order to find similar-minded objects. The travel schedule is extracted by computing *Object to Cell Similarity*.

To visit a cell, an object register itself on the *Cell Manager*. The Cell Manager handles a directory of all registered objects through their addresses. The *Relationship Manager* is responsible for computing the similarity between objects, and for adopting human-centered behavior, discussed in the next section.

3.4 Owner Control

SIoT Function. The owner control refers to specific policies put forward by the owner to govern the interactions of objects, e.g., what information to share, and which connections to establish. In our running example, the owner may decide to not allow the smart thermostat to publicly publish to the internet information from its profile, e.g., configuration, sensor readings, household composition, habitation patterns, but may allow the smart thermostat to connect and exchange information with other objects from the same manufacturer.

Implementation in DANOS. DANOS goes beyond owner control and dictates how objects should function under the policies put forth by their owners. DANOS follows a anthropomorphic design, and enables objects to inherit human factors from its owner and act accordingly. Our assumption is that an anthropomorphic behavior of the objects would help the owner to better understand the implications of the selected control profile [32], and would positively affect the objects' relationship establishments, the network formation and the information quality, aspects that are evaluated in Section 5.

Objects in DANOS adopt a *human-centered behavior* (HCB), which is introduced in [32] after consulting research outcomes and lessons learned from human social networks and personality traits, and is outlined in Table 2. HCB is driven by the user specifics (as a rule-based set of actions), and defines the different behaviors of objects during interaction by manipulating the similarity metrics, and generating the different opinions that might have for each other during their connection. HCB consists of five distinct behaviors, approaching, helpfulness, attraction, risk-taking, and risk-avoiding, and is specified by a directed subjective opinion of one object towards another. The HCB provides the opportunity to configure the friendship establishment process from an object's perspective. In brief, *Helpfulness* defines a behavior where objects perceive an increased similarity towards newly on-boarded objects to avoid information clusters. *Attraction* means that an object will be perceived by others as being more similar to them. *Risk-Taking* and *Risk-Avoiding* behaviors alter the perceived similarity of other objects.

In our running example, the thermostat owner might have high neuroticism and low openness (personality traits that are part of the object profile), which means that they might appreciate a fixed pattern in temperature adjustments (as according to their behavior, they are not always so keen to tackle new

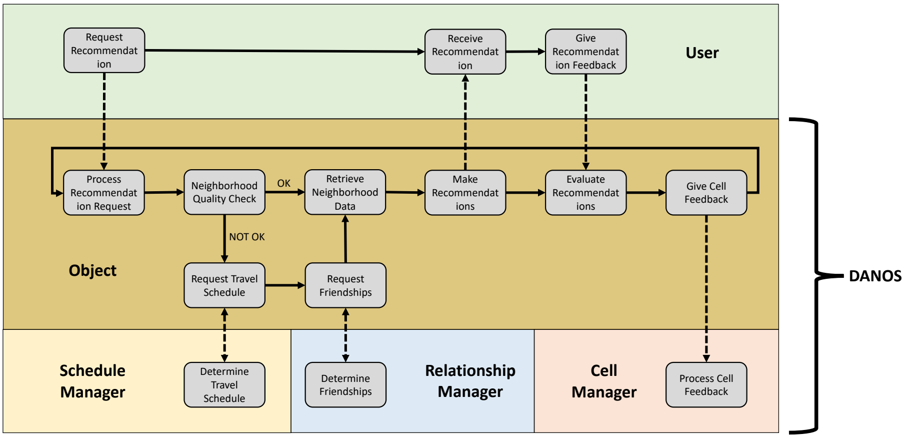


Fig. 3: Recommendation Workflow in DANOS

challenges and might experience mood swings or high levels of anxiety in more unstable conditions). The thermostat thus adopts a risk-avoiding behavior (see Table 2) leading the object to avoiding connecting with objects that are not highly similar to itself.

4 Recommendations in DANOS

In this section, we discuss how recommendations are provided in DANOS. We start by overviewing the workflow using our running example, and then discuss in detail how the workflow is executed.

4.1 Overview

We overview the recommendation workflow in DANOS, presented in Figure 3, by explaining how the running example, presented in Section 1, adapts to DANOS.

We start by explaining the workflow from the viewpoint of the user, the owner of the smart thermostat, which is captured at the top of Figure 3. The owner *requests a recommendation*, which in our example is a continuous request for temperature settings. This request is delegated to the social agent of the thermostat, and eventually the user *receives a recommendation* from the smart object; here, the recommendation is a temperature setting. The user may or may not *give recommendation feedback* to the smart object. In the running example, feedback is observed when the user manually adjusts the temperature to a different setting than what the smart thermostat recommended. The recommendation feedback helps the smart thermostat make better temperature recommendations in the future.

Let us now discuss the workflow from the viewpoint of the various components of DANOS. Initially, the object is ready to *process a recommendation request*. At this point, the object has the following information. It knows about the *object specifics*, such as information about the current state and context collected from other objects in the home IoT network (e.g., date, time, weather, home state, energy consumption). Moreover, the object knows *user specifics*, such as the owner's profile (e.g., age, personality characteristics like strong introvert, low in openness to experience and high levels in the neuroticism scale), preferences (e.g., prefers a very warm house in the morning, looks to aggressively minimize energy consumption), and feedback (e.g., examples when the recommended temperature was different than the desired). Further, based on its owner's personality traits, the object adopts a human-centered behavior. Specifically, (the social agent representing) the smart thermostat is not that keen to approach other objects, and has a tendency towards a risk avoiding behavior, inheriting from its owner's introversion — see Table 2, for details regarding how specific personality types are translated into social agent behaviors. The smart thermostat has also *interaction specifics*, which include the neighborhood of friend thermostats with whom it has established connections in the past, any information the friends have shared, and quality measures of the utility of these friends.

Upon receiving a recommendation request, the thermostat first does a *neighborhood quality check* to determine whether the collected information available locally is sufficient to process the request. If the check returns OK, the request is processed without going into the DANOS cyberspace. The thermostat then needs to update its information from its friends, and thus *retrieves neighborhood data*. The next step is to utilize all this information in order to *make recommendations* and propose a temperature setting to its owner. Based on the owner's feedback, the object *evaluates the recommendations* and updates local model parameters. Finally, the object *gives cell feedback* to appropriate cells in DANOS.

In case the neighborhood quality check returns NOT OK, the object need to visit the DANOS cyberspace in order to establish new connections to improve the neighborhood quality. Thus, the object *requests a travel schedule* from the scheduling manager. The latter then examines the cells available for the given intent (temperature adjustement) and *determines the best travel schedule* for the thermostat's social agent. The schedule is a series of cells to visit, and at each cell the object *requests friendships* from the cell's relationship manager. The manager *determines friendships*, which is a list of other objects that are best suited to provide useful information to the thermostat object in its mission to make good temperature adjustments. After the owner gives their feedback to the thermostat, the thermostat in turns gives cell feedback to the cell manager, who *processes cell feedback*.

In what follows, we discuss in detail the important steps of the recommendation workflow in DANOS.

4.2 Recommendation Workflow

4.2.1 Notation

In the following, we refer to a user (owner of an object) and its object interchangeably; we use the former in the context of recommendations, and the latter in the context of the SIoT network. A user u has a *profile* abstractly represented by the vector p_u. The profile contains the object specifics (e.g., device characteristics), the interaction specifics (e.g., neighborhood), and the user specifics (e.g., personality traits) except preferences and feedback/ratings, which are considered separately.

We assume a specific intent for the SIoT objects: recommendations of *items* from a set I. We assume that an item $i \in I$ has a *content* (e.g., product categories) that is abstractly represented by the vector c_i. For example, the k-th dimension $c_i[k]$ of the content vector may indicate the membership degree of item i to the k-th product category.

A user u provides a *rating* (feedback) to an item i, which is represented as $r_{u,i}$ and is normalized in the $[0,1]$ range. We use I_u to denote the subset of items that user u has rated. The feedback given by a user determines the user's *preference* on item content, represented by the vector π_u. Specifically, the k-th dimension of the preference vector indicates the inclination of the user towards the k-the content aspect, and is thus computed as:

$$\pi_u[k] = \sum_{i \in I_u} c_i[k] \cdot r_{u,i}.$$

Given two users u, v, we define three similarity values, based on their profiles, their ratings, and their preferences.

The *profile similarity* between two users u, v is denoted as $s_p(u,v)$ and is computed as the cosine similarity of their profile vectors:

$$s_p(u,v) = \frac{\langle p_u, p_v \rangle}{||p_u|| ||p_v||}.$$

Similarly, their *preference similarity* is the cosine similarity of their preference vectors:

$$s_\pi(u,v) = \frac{\langle \pi_u, \pi_v \rangle}{||\pi_u|| ||\pi_v||}.$$

The *rating similarity* between two users u, v is denoted as $s_r(u,v)$ and is computed as the adjusted Pearson Correlation Coefficient of the users' ratings normalized to $[0,1]$:

$$s_r(u,v) = \frac{1}{2} \left(\frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{v,i} - \overline{r}_v)^2}} + 1 \right),$$

where $\overline{r}_u$ is the mean rating of user u.

4.2.2 Neighborhood Quality Check

We first discuss what is the neighborhood of an object and present a single metric to capture its quality. Then, we explain how an object decides to cancel friendships and how it determines if its neighborhood has sufficient quality.

Friend Neighborhood

An object can only exchange information directly with other objects it has established a friendship with. In the context of recommendations, the information exchanged concerns the items, i.e., item content and ratings. Therefore, an object has only a limited view on the catalog I of items: it only knows what has been shared from its friends. Consequently, to be able to make good recommendations, an object needs to establish and maintain meaningful friendships.

At any point in time, a user u has a set of friends denoted as F_u. The information available to u consists of the content and ratings of all items rated by a friend. Based on this information, user u computes the preference $s_\pi(u, v)$ and rating $s_r(u, v)$ similarity to each friend v. Moreover, when u established the connection with v in DANOS, u received the profile similarity $s_p(u, v)$ from the relationship manager — the profile vector p_v is considered private and is not shared.

So, for each of its friends, user u knows its preference, rating, and profile similarity. These similarity values are aggregated into an non-symmetric overall similarity. Specifically, the *overall similarity* of v to user u in the context of u's network is the convex combination of the three similarities normalized among all friends, and is computed as:

$$s(u, v;\ F_u) = w_p \cdot s_p(u, v;\ F_u) + w_r \cdot s_r(u, v;\ F_u) + w_\pi \cdot s_\pi(u, v;\ F_u), \quad (1)$$

where the similarity weights w_p, w_r, w_π are in $[0, 1]$ and sum to 1; and each normalized similarity s_x for $x \in \{p, r, \pi\}$ is computed relative to the friends F_u of u as $s_x(u, v;\ F_u) = \frac{s_x(u,v)}{\sum_{v' \in F_u} s_x(u,v')}$. The similarity weights are model parameters whose values are learned over time in order to improve recommendation effectiveness (see Section 4.2.4).

For making recommendations, a user would want to be friends with users that are similar-minded, i.e., have high profile, rating, and preference similarity. However, having a highly similar circle of friends may cause a *filter bubble*, reducing the number of items the recommender may choose from (catalog coverage). Thus, to build a useful friendship network, the user should also consider the diversity of preferences among its friends.

We define the *preference diversity* of a friend v with respect to the friendship network of u as the average preference dissimilarity of v to every other friend of u:

$$d(v;\ F_u) = \frac{1}{|F_u \setminus \{v\}|} \sum_{v' \in F_u \setminus \{v\}} (1 - s_\pi(v, v')),$$

where v is excluded when computing the average. Intuitively, preference diversity quantifies how dissimilar a friend is with respect to the other friends of a user.

Combining the overall similarity of a friend v to user u and the preference diversity of friend v, we derive the *quality of friendship* (QoF) of friend v to user u. Similar to the MMR [96] and xQuAD [97] frameworks, we compute QoF as:

$$q(v;\ F_u) = (1 - \lambda) \cdot s(u, v;\ F_u) + \lambda \cdot d(v;\ F_u),$$

where the two terms are traded off by an external parameter λ. Intuitively, a friend v has a high QoF w.r.t. to u, if v is similar to u and/or if v's preferences are dissimilar to those of u' other friends.

Checking Neighborhood Quality

As users' preferences and rating history changes over time, so does the quality of friendship. A user should try to maintain a circle of friends that have high quality. Therefore, it may decide to cancel some friendship connections. We employ a simple outlier detection mechanism to cancel a friendship in the network of u. Consider the distribution of QoF values $q(\cdot;\ F_u)$ among the friends F_u. Let Q_ϕ denote the $1/\phi$-quantile of this distribution; for example when $\phi = 0.25$, we get the first quartile. Then define the difference $\Delta = Q_{1-\phi} - Q_\phi$; again when $\phi = 0.25$, this is the interquartile range. Then, we cancel friendship with any friend that has a QoF below $Q_\phi - \Delta$, i.e., is a negative outlier.

Each object has a target number of friends. When an object has few friends, its neighborhood quality essentially decreases. The object has a simple probabilistic mechanism to determine whether its neighborhood quality is sufficient. If the object is missing a ratio p of the target number of friends, the neighborhood check returns NOT OK with a probability of p.

4.2.3 Make Recommendations

In DANOS, an object only uses information from itself and its friends to process a recommendation request. Specifically, we assume that the object has collected the content and ratings of all items rated by each of its friends. The recommender thus processes this information to derive its recommendations. Any collaborative filtering or content-based approach, or some combination thereof, can be used, and the selection is orthogonal to DANOS.

For evaluation purposes, we implement a simple hybrid user-neighborhood collaborative filtering method that also takes into account item content similarity. Specifically, we predict the rating of each item $i \in \cup_{v \in F_u} I_v$ known to the user u as:

$$\hat{r}_{u,i} = \overline{r}_u + \sum_{v \in F_u} s(u, v;\ F_u) \cdot (r_{v,i} - \overline{r}_v), \tag{2}$$

since the similarities are normalized, i.e., $\sum_{v \in F_u} s(u, v;\ F_u) = 1$.

4.2.4 Evaluate Recommendations

The object presents a list of recommendations to the user and receives from them their feedback. In the most general setting, the feedback is implicit and indicates the item the user interacted with. The object can then evaluate the recommendation it presented. Specifically, it computes the *reciprocal rank* (RR) of the interacted item in the list. If the interacted item was in position k, the RR is $1/k$. The higher the RR the more effective the recommender is.

Based on this feedback, DANOS improves the recommendation effectiveness by seeking to create better neighborhoods for each object. It achieves that in an indirect and a direct way. Indirectly, the new feedback will change the preferences and feedback of the object. This in turn will change its preference and rating similarities with other objects. This means that in the future, objects will have better information to decide how to connect to each other.

More crucially, in a direct way, the user feedback will affect the values of the weights used to compute the various similarity scores and the quality of friendship. Specifically, the object runs an *online learning process* to discover the optimal values of w_p, w_r, w_π.

After each feedback, the object tries six different value configurations for the similarity weights, and determines the configuration with the highest RR. Since the three weights are tied together to sum to one, it suffices to define the configuration values for two of them, say w_p, w_r. In a configuration, each of these two weights will have three options: remain the same, increase by η, or decrease by η, where η is called the *learning rate*. The configuration that achieves the highest RR for the current feedback round is called the winner. After τ feedback rounds, where τ is called the *learning period*, the object identifies the configuration with the most wins. It then applies the overall winner configuration on the current weight values, i.e., it increases/decreases their values by τ, or keeps them unchanged.

4.2.5 Give Cell Feedback

As discussed, the user feedback will affect the object's preference and feedback information, and ultimately the QoF of each friend. The object identifies the most useful of its friends, with a process similar to finding QoF outliers as discussed for friendship cancelling, only this time positive outliers (with QoF above $Q_\phi - \Delta$) are selected. For each such useful friend, the object recalls the cell where they established connection, and sends a *cell feedback*. This feedback consists of the object's and its friend current profile, preference and rating information, as well as the value of the evaluation metric (RR) determined from the user rating/feedback.

4.2.6 Process Cell Feedback

Recall that each cell in DANOS serves as a "meeting place" for objects. The effectiveness of the distributed SIoT environment depends on how well an object can identify like-minded objects. This implies that DANOS cells must become specialized enough with a purpose, e.g., this cell is frequented by objects that tend to like a specific type of items. To support this specialization, a cell receives cell feedback to determine whether the object connections established within it were meaningful for the objects involved.

A cell maintains a fixed-size memory of object profiles from the recently received cell feedback. After receiving a cell feedback, the two associated object profiles are inserted in the cell memory, while the two oldest object profiles are evicted. The cell then computes a *cell proxy* that is the RR-weighted average of the object profiles in the cell memory. In essence, the cell proxy plays the role of a virtual object profile that is representative of all objects that have established meaningful connections in this cell. The cell proxy plays an important role in determining a travel schedule for objects in DANOS.

4.2.7 Determine Travel Schedule

The steps for processing a request for travel schedule are shown in Figure 4. Given the intent, the object initially requests from the *Intent Manager* the appropriate area (e.g., that handles movie recommendations) (step 1). Through the returned area address, the object is now able to request from the area's *Schedule Manager* a travel schedule. This schedule consists of a list of cells that the object should travel to in order to maximize the chances of finding similar objects (step 2). To compute the schedule, the Schedule Manager computes the similarity between the object's profile information and each cell proxy. Recall that the cell proxy is identical to the profile of a virtual object, so Eq. 1 is used to compute the similarity between an object and a cell. Then, the travel schedule contains all cells are ordered from most to less similar (step 3).

Given a schedule, the object visits each cell in turn, looking to establish new friend connections. The traveling terminates as soon as the object has enough (i.e., up to some maximum predefined number) connections.

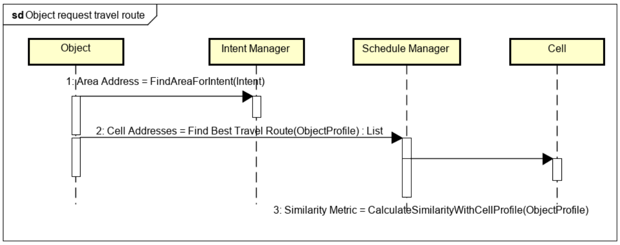


Fig. 4: Processing a Travel Schedule Request

4.2.8 Determine Friendships

The steps for traveling to a cell and for processing friendship requests are shown in Figure 5. First, the object initiates the travel by contacting the area

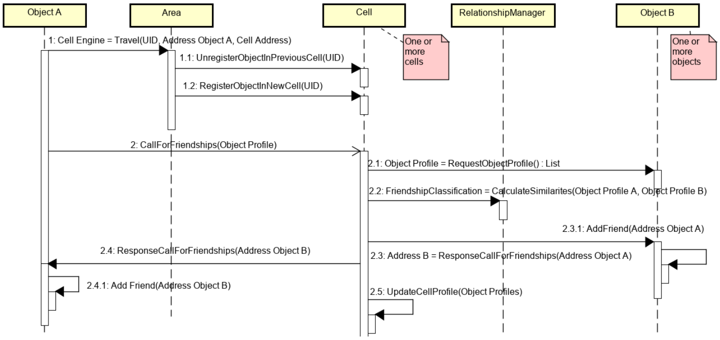


Fig. 5: Object Traveling to, Requesting Friendships in, and Providing Feedback to a Cell

manager (step 1). Then, the manager unregisters the object from its current cell (if any) (step 1.1) and registers it in the desired cell (step 1.2).

Once in a cell, the object requests for friendships (step 2). The cell first requests from all objects to send it their object profiles (step 2.1). The cell then forwards the object's and the collected profiles to the relationship manager, who calculates the overall similarity of the object to each other object registered in the cell (step 2.2). The manager compiles a list of the most similar objects. For each target object in the list, the cell forwards the address of the requesting object (step 2.3). Each target object adds the requesting object to its neighborhood (step 2.3.1). Similarly, the address of each target object is sent to the requesting object (step 2.4), who in turns adds them to its neighborhood (step 2.4.1). Figure 5 also depicts the cell feedback processing that results in the updating of the cell proxy (step 2.5) as discussed in Section 4.2.6.

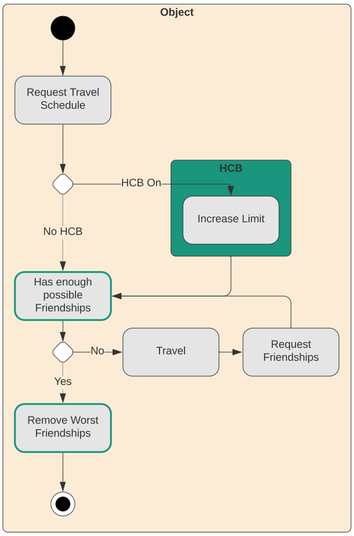
4.2.9 Human Centered Behavior

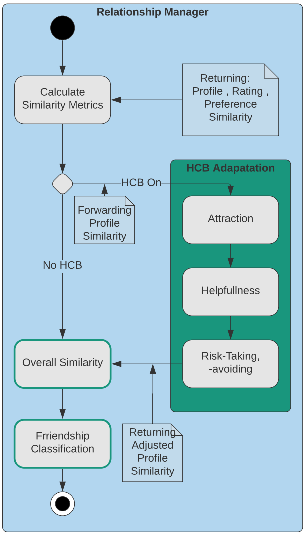
Figure 6 presents the process flow when HCB is on and when off, from the perspective of the object and the relationship manager. As discussed, HCB essentially dictates how much the opinion of an object changes. The behavior changes are presented in the following.

Approaching

The object itself processes this behavior. When HCB is on, the maximum number of friends the object can have is increased up to two times. The intensity of human-center behavior is controlled by a single external parameter, denoted as γ, that takes values between 0 (no HCB) and 1. If $max\_num\_friends$ denotes the maximum number of friends, then HCB changes this value as:

$$max\_num\_friends \leftarrow (1+\gamma) \cdot max\_num\_friends.$$





(a) HCB Processing in the Object (b) HCB Processing in the Relationship Manager

Fig. 6: Human-Centered Behavior in DANOS

The other four behaviors are processed by the relationship manager. If HCB is on, the value of the overall similarity between two objects will change. If multiple behaviors apply, they have a cascading effect.

Helpfulness

If the object u exhibits this behavior, it will be attracted more to another object v. In this case, the overall similarity is adjusted as:

$$s(u, v) \leftarrow 1 - (1 - s(u, v))^2.$$

Attraction

An object u exhibiting this behavior wants to connect more to an object v that has fewer friends than average. Let ρ denote the ratio between the number of friends v has and the average number of friends (as observed in the cell over time); ρ is less than 1. Then, the overall similarity between objects u and v is adjusted as:

$$s(u, v) \leftarrow 1 - (1 - \rho)^2 \cdot (1 - s(u, v))^2.$$

Risk-Taking and Risk-Avoiding

In decision theory, risk-taking or risk-avoiding behavior is modeled by applying a convex or a concave, respectively function to the utility. In our context, when an object u is risk-taking, we adjust its similarity to another object v as:

$$s(u, v) \leftarrow 1 - (1 - s(u, v))^2.$$

When u is risk-avoiding, we adjust its similarity to another object v as:

$$s(u, v) \leftarrow s(u, v)^2.$$

5 Evaluation

In this section, we present a detailed experimental evaluation of the DANOS architecture, seeking to investigate the research questions posed in Section 1. Specifically, for each question we formulate concrete hypotheses that we then test.

RQ₁ How good are the neighborhoods formed in DANOS? (*Dynamicity*)
- ***H₁*** The online learning process for neighborhood formation converges quickly.

RQ₂ How efficient is the discovery mechanism of DANOS? (*Decentralization*)
- ***H₂*** Increasing the neighborhood size has diminishing marginal gains.
- ***H₃*** Cell specialization can be achieved at various decentralization levels.

RQ₃ What is the effect of the human-centered behavior in DANOS? (*Anthropomorphism*)
- ***H₄*** The HCB leads to more effective recommendations and increased cell specialization.

RQ₄ How effective is DANOS in processing IoT recommendations? (*Effectiveness*)
- ***H₅*** DANOS makes as effective recommendations as those of a centralized recommender.
- ***H₆*** The dynamicity enables DANOS to makes more effective recommendations than those in a static network.

We next present the evaluation setup (Section 5.1), followed by the investigation of the four research questions (Section 5.2–5.4), and conclude with limitations of our study (Section 5.6).

5.1 Setup

We present a setup that allow us to investigate in depth the aforementioned hypotheses. While DANOS is a fully functional system, it has not been deployed in a real IoT environment. Therefore, we seek to create a realistic simulation that would allow us to properly explore all theoretical and technical perspectives of DANOS. Please refer to Section 5.6 for the limitations of our evaluation.

Dataset. Without a real-life deployment of DANOS, we look for datasets that would enable a simulation. Specifically, we require a dataset that logs multiple recommendation requests and also contains user feedback. Due to the lack of such a dataset in the IoT domain, to our knowledge, we turn to a standard dataset in the recommender systems community, the MovieLens 1M dataset[1]. This dataset contains 1 million ratings on a five-star scale from 6 thousand users on 4 thousand movies. The dataset includes demographics data for each user, and genres for each movies. The dataset however contains no personality traits. To address this, we enrich the data by leveraging the work of [98] that studies the connection between personality traits and movie genres. Specifically, for each movie genre and gender, the authors present the average (stereotype) personality traits of the user that is interested in this genre. We thus use these stereotypes to assign personality traits to users. For each user, we first identify the movies they have enjoyed (rated with 4 of 5 stars) and construct the user's genre mix (e.g., the user likes 30% action, 70% adventure movies). Then, to each user we assign personality traits that are computed as the weighted average of the stereotypes associated with the genres they like (e.g., 30% action stereotype and 70% adventure stereotype).

SIoT Simulation. For the SIoT simulation, we randomly select 800 users from the MovieLens dataset. We experiment with various user-based samples, and all tested samples lead to similar findings. Each user is associated with an SIoT object. Initially, each object inherits the profile of its user, and joins DANOS to make connections based on its profile similarity. After this bootstrapping phase, we scan the given user ratings, treating each as a request for recommendation.

DANOS configuration. We define a single area in DANOS for handling movie recommendations. We vary the number of cells from 1 up to 32, to investigate different decentralization levels. We vary the neighborhood size (maximum number of friends) of objects from 20 up to 50 — the actually value for each object may vary because of HCB. Initially, all objects have equal similarity weights ($w_p = w_r = w_\pi = 1/3$).

Baselines. We compare the recommendation strategy of DANOS with two baselines. The first, called *Central*, is a centralized recommender system that has access to all data. The purpose of this baseline is to investigate H_4 and see whether the distributed environment and the profile specialization within cells can substitute for the missing global knowledge. The second baseline, called *Static*, is a restricted version of the DANOS recommender in that the objects' network is created once during the bootstrapping phase, and remains fixed ever since. Its purpose is to investigate H_5, and see whether the dynamic network evolves effectively over time.

To keep the comparison fair, the baselines and our SIoT recommender use the same underlying recommendation strategy, the hybrid user neighborhood-based approach. The method only differ in what data they have access to. In Central, each user/object has access to all data. In Static and in our approach, an object can only access the data that its friends have, which essentially depends on the structure of the objects' social network formed.

---

[1] https://grouplens.org/datasets/movielens/1m/

Evaluation Metrics. We care primarily for the effectiveness of the recommender. We measure prediction accuracy in terms of the root mean squared error (*RMSE*) of the rating predictions made by the recommenders. Moreover, we care for ranking accuracy, and use the positive ratings (with score 4 or 5) to measure the mean reciprocal rank (*MRR*) and *Hit Ratio*. Moreover, we measure the *List Size*, the number of items that can be recommended at any point.

We also care about the quality of the neighborhoods formed in DANOS. For the neighborhood of each object, we measure the three (profile, preference, and rating) similarity measures, the overall similarity, the mean preference diversity, and the mean quality of friendship, as defined in Section 4.2.2.

Going at the level of cells, we care about their specialization. For each cell we measure the mean distance in the object profiles stored in the cell memory, and distinguish between mean distance in profiles, ratings, and preferences.

At the network level, we count specific events: the number of cell visits (Num-Cell-Visits), the number of travel schedules requested (Num-Schedule-Requests), the number of friendships requested (Num-Friend-Requests), and the number of travels to cells (Num-Travels).

Object-level metrics are averaged across all objects. Cell-level metrics are averaged across cells. For each evaluation metric, we report its mean value along the entire simulation, and also report its rolling mean at each time point in the simulation.

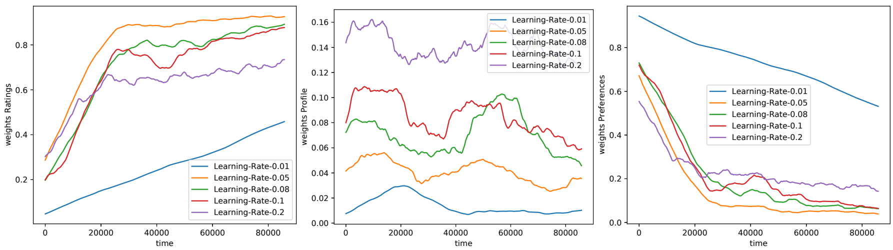
5.2 RQ_1: Dynamicity

We start by studying the dynamicity of DANOS, and investigate the hypothesis that the online learning process for neighborhood formation converges quickly (H_1). There are two hyperparameters that control the learning process, the learning rate and the learning period. We experiment with various values of these hyperparameters. We fix the number of DANOS cells to 1 for this round z of experiments.

Learning Rate.

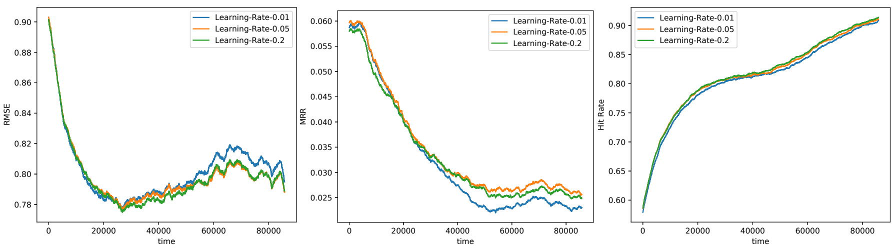
We fix the learning period to 10, and vary the learning rate in the values 0.01, 0.05, 0.08, 0.1, 0.2. We then measure the various recommendation effectiveness metrics. Table 3 presents the mean values of the metrics over the entire simulation. We observe that 0.1 is the best setting in terms of RMSE and Hit Rate, while 0.05 is the best for MRR and List Size. We consider MRR to be the most important metric, and thus we choose 0.05 as the optimal learning rate.

Figure 7 shows the convergence of the learning process over time. We observe an interesting behavior. Over time, the rating similarity weight increases, while the preference similarity weight decreases. This means that as the object collect more feedback from their users, they tend to rely more on that piece of information, rather than the more abstract type of information provided by genre preferences. Moreover, we observe that the weight for profile similarity remains low but fixed over time around 0.10. This indicates that



(a) Ratings weight (b) Profile weight (c) Preferences weight

Fig. 7: Convergence of Learning Process For Different Learning Rates



(a) RMSE (b) MRR (c) Hit Rate

Fig. 8: Recommender Effectiveness for Different Learning Rates

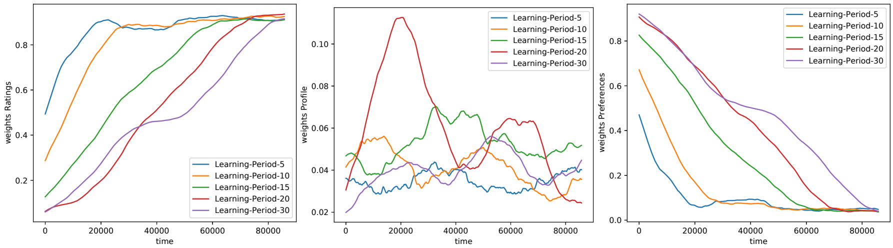
profile similarity plays a smaller but not negligible role in recommender effectiveness. Regarding the convergence of different learning rates, we not very distinct behaviors in convergence. The learning rate of 0.05 converges faster to a steady state. On the other hand the small rate of 0.01 is unable to reach its steady state.

Figure 8 presents the recommender effectivess for various learning rates. In agreement with Table 3, we observe that the rate of 0.05 achieves the best effectiveness (has low RMSE, high MRR, high Hit Rate), although the margins are not that wide.

Learning Period.

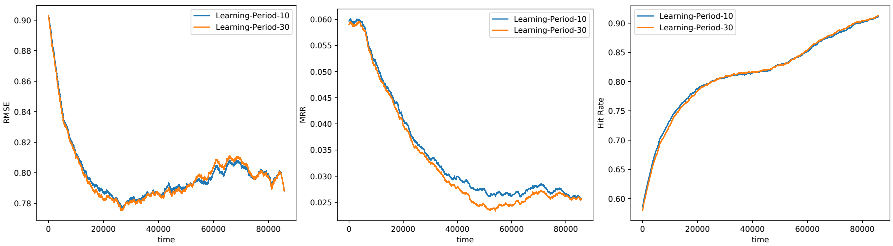
We proceed by fixing the learning rate to 0.05, and vary the learning period in the values 5, 10, 15, 20, 30. We measure recommendation effectiveness, and report the mean values of the metrics over the entire simulation in Table 4. We find that the value of 15 achieves best RMSE, 10 achieves best MRR and Hit Rate, while 5 has the largest List Size. As we primarily care for MRR, we conclude that the learning period of 10 is the optimal.

Figure 9 shows the convergence of the learning process for the learning periods tested.We observe that the ratings similarity weight increases, while



(a) Ratings weight (b) Profile weight (c) Preferences weight

Fig. 9: Convergence of Learning Process For Different Learning Periods



(a) RMSE (b) MRR (c) Hit Rate

Fig. 10: Recommender Effectiveness for Different Learning Periods

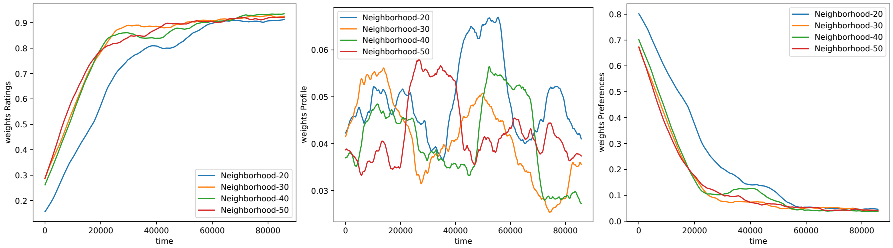
the preference similarity weight decreases over time. Smaller learning periods appear to lead to convergence faster.

Figure 10 shows the recommender effectiveness for the various learning periods. The margins are small, but the same conclusion as in Table 4 applies, learning period of 10 leads to more effective recommendations. In what follows, we keep fixed the learning rate at 0.05 and the learning period to 10.

In this section, we investigate the effect of the decentralization approach of DANOS in terms of task effectiveness. Specifically, we study the effect of the two main decentralization parameters, locally at each object (the neighborhood size), and globally at DANOS (the number of cells per area, a.k.a. the decentralization level), as captured by hypotheses H_2 and H_3.

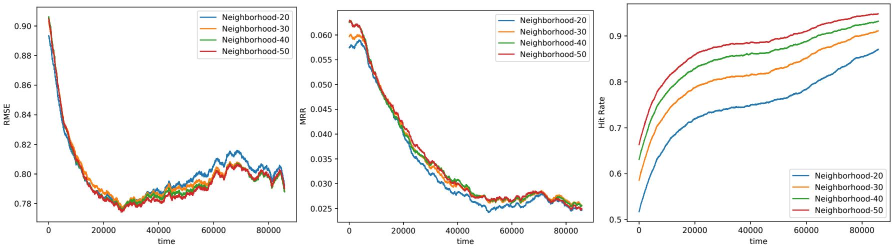
5.2.1 Local Decentralization Level

The size of an object's neighborhood affects the amount of data directly available based upon recommendations are generated. Clearly, the larger the neighborhood size, the more effective the recommendations can become. The hypothesis we investigate is whether there is diminishing returns situation, as stated in H_2. Therefore, we vary the maximum number of friends an object



(a) Ratings weight (b) Profile weight (c) Preferences weight

Fig. 11: Convergence of Learning Process For Different Neighborhood Sizes



(a) RMSE (b) MRR (c) Hit Rate

Fig. 12: Recommender Effectiveness for Different Neighborhood Sizes

can have among the values of 20, 30, 40, and 50. We report recommender effectiveness in Table 5. There is a clear trend, increasing the size leads to more effective recommendations.

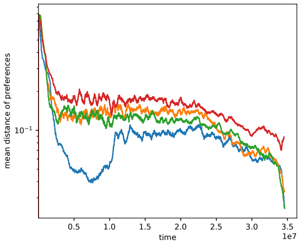
Figure 11 presents the convergence of the learning process for the various neighborhood sizes. Overall, the process converges at the same rate for all sizes. The convergence is slightly slower when the neighborhood is set to 20.

Figure 12 shows recommendation effectiveness over time for the different neighborhood sizes. Higher neighborhood sizes lead to better recommendations in all metrics, with the differences being more pronounced in the case of the Hit Rate.

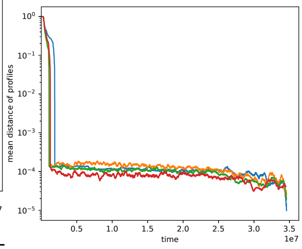
5.2.2 Global Decentralization Level

We now investigate the effect of the global decentralization level, i.e., the number of cells per area. We expect that the increased decentralization of the network would enable higher specialization of the cells in time with respect to different metrics. To study this effect, we vary the number of cells between 4, 8, 32.

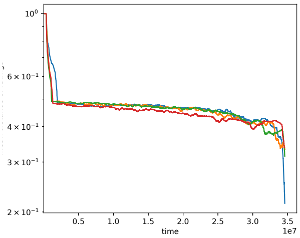
With each cell size, we want to capture the ability of cells to specialize, i.e., how good are the object profile clusters in each cell. For this, we measure the mean distances for each object profile dimension (rating, preferences, and



(a) Mean Preference Distance

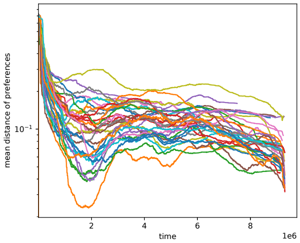


(b) Mean Profile Distance

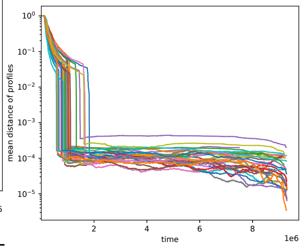


(c) Mean Rating Distance

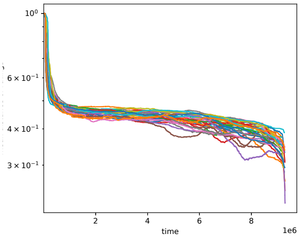
Fig. 13: Cell Specialization Over Time for 4 Cells



(a) Mean Preference Distance



(b) Mean Profile Distance



(c) Mean Rating Distance

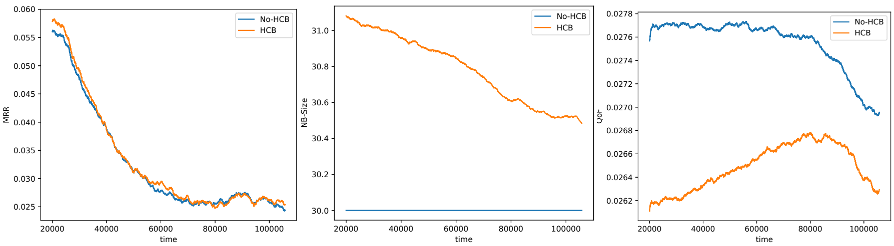
Fig. 14: Cell Specialization Over Time for 32 Cells

profile). Table 6 shows the mean distances at each level of decentralization. Specialization occurs when the mean distances decrease as the number of cells increase. We observe specialization in terms of rating, where the mean distance decreases ($4 \rightarrow 8$: -0.92%, $8 \rightarrow 32$: -4.86%), and in terms of preferences, where the mean distance decreases ($4 \rightarrow 8$: -5.99%, $8 \rightarrow 32$: -1.81%), but not in terms of profiles.

Figures 13 and 14 depict the specialization over time for 4 and 32 cells, respectively. Each line in the figures corresponds to a different cell. In both sets of figures, we observe that the means decrease over time, and thus the cells specialize over time.

5.3 RQ_3: Anthropomorphism

In what follows, we investigate the effect of human-centered behavior. When objects adopt the various interaction behaviors (as suggested in Table 2) while operating in the SIoT network, we hypothesize that key performance metrics would be positively influenced. For the sake of completeness, we also perform a further evaluation of the distinctive behaviours of objects, based on their personality traits (and theoretical perspectives of each one), verifying their



(a) MRR (b) Neighborhood Size (c) QoF

Fig. 15: HCB vs No HCB

operation in relation to their outcomes and expected impact. In this round of experiments, we set the number of cells to 32.

Table 7 shows how many objects are strong and how many are weak in each personality trait. Table 8 shows how many objects adopt the individual behaviors.

5.3.1 Overall Effect

First, we want to see how HCB affects DANOS in general, and investigate hypothesis H_4. Table 9 and Figure 15 present a comparison when HCB is used and when not. With HCB, the MRR increases by 2.16%, while objects tend have a slightly larger neighborhood size (2.61%). The mean quality of friendship decreases (-4.42%) along with preference diversity (-3.92%), while preference similarity increases (0.78%). All network performance metrics except Num-Cell-Visits increase: Num-Schedule-Requests (13.64%), Num-Friend-Requests (17.87%), Num-Cell-Visits (-1.45%) and Num-Travels (22.06%). Figure 15 shows that HCB has the most effect in the beginning of our simulation, where preferences are the most important source for improving MRR.

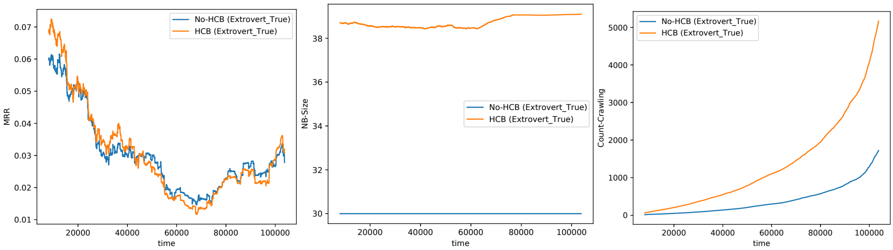
To investigate cell specialization with HCB, we present the results in Table 10, which are to be contrasted with those in Table 6. With HCB, the specialization is better, as mean distance decreases more in profiles (-64.15%), in ratings (-0.42%) and in preferences (-0.53%) (suggesting the overall acceptance of H_4).

5.3.2 Effect of Individual Behaviors

After observing the overall effect of HCB, we focus on individual behaviors and how they affect objects exhibiting them.

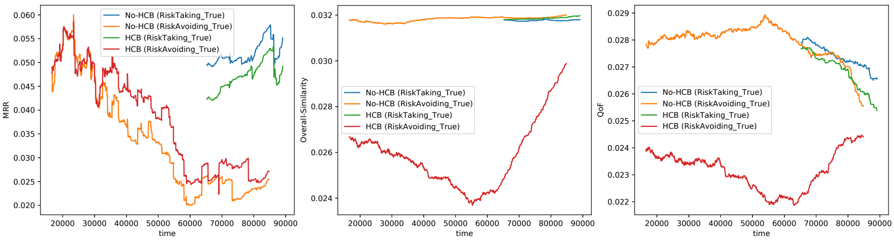
Approaching.

A highlighted behavior of the extrovert dimension is that objects are more expressive and keen to make friends. For evaluating this behavior, we isolate the objects that are strong extroverts (87 objects) and we compare them with



(a) MRR (b) Neighborhood Size (c) Count Crawling

Fig. 16: Approaching



(a) MRR (b) Overall-Similarity (c) QoF

Fig. 17: Risk-Taking and Risk-Avoiding

a simulation execution with the same objects where HCB is off (not applied). The results are presented in Table 11 and in Figure 16.

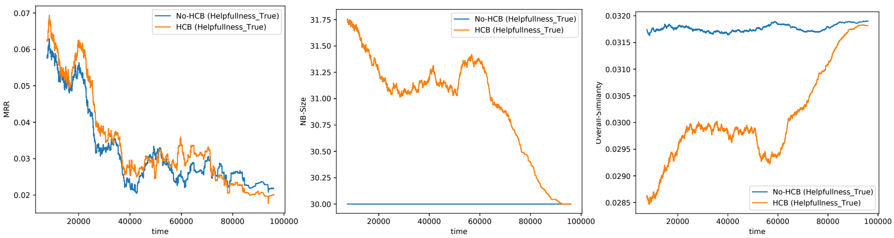
The results show, that the MRR increases by 3.09% with a larger Neighbourhood Size (28.92%) and a decreased QoF (-39.72%). The Preference Diversity increases mostly (33.04%) with a decrease in Preference Similarity (-1.98%). All network parameters increase; Friendship Crawling (246.42%), Num-Friendship-Canceling (37.86%), Visited-Cells (11.65%), and Num-Travels (32.82%).

Through the increasing MRR, the object resource costs are a bigger Neighborhood Size and a decreased Quality of friendship. By interpreting this to a human network, an extrovert person will have more friends with a higher diversity of preferences, but a lower quality of friendships.

The approaching objects generate high network traffic through more friendship requests, more traveling, and visiting more different cells. By interpreting this to a human network, an extrovert person travels more and visits more different places and gets to know more people.

Risk-Taking and Risk-Avoiding.

To evaluate this behavior, we isolate (by subdividing the evaluation data), the objects that are risk-taking (7 objects), and risk-avoiding (17 objects) and



(a) MRR (b) Neighborhood size (c) Overall-Similarity

Fig. 18: Helpfullness

we compare them with HCB off. The results are shown in Table 12 and in Figure 17.

For the risk-taking objects, the results show, that the MRR decreases by 8.69% with a higher Overall Similarity (0.31%) towards the other objects, a decreased QoF and Preference Diversity (-2%). The profile similarities and Num-Crawling are less affected (<1%).

The interpretation for the risk-taking behavior, as shown by a decreased MRR and a low QoF, is that objects are more open to experiences, by connecting to other objects at the expense of recommendation effectiveness.

For the risk-avoiding objects, the results show, that the MRR increases by 10.66% with a lower Overall Similarity (-19.18%) towards the other objects. The QoF decreases (-17.14%) as well as Preference-Diversity (-1.76%). The profile similarities change less (<1%), while the Num-Crawling strongly increases (81.59%).

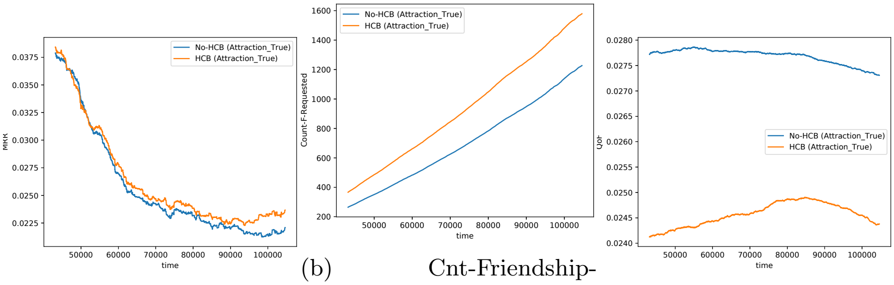
The risk-avoiding behavior leads to an improved recommendation effectiveness (MRR) but with decreased Overall-Similarity and QoF. A risk-avoiding object goes only into a friendship if the similarity is really high. To achieve this, the object decreases its perceived similarity towards other objects and gets fewer friends from a cell, while it needs to travel more to reach the desired number of friendships.

Observing the risk-taking and risk-avoiding objects in Figure 17, the risk-avoiding (17) objects are mostly active the whole time, whereas the risk-taking (7) are mostly active at the end.

Helpfulness.

The results, presented in Table 13 and Figure 18, show that MRR increases by 9.44% with a larger Neighbourhood Size (3.77%) and a decrease in Overall Similarity (-6.29%), QoF (-5.42%) and Preference Diversity (-3.46%). For the network metrics, the number of crawling increases (22.63%) and the Num-Friendship-Requests (20.78%) with fewer visited cells (-2.06%) and a higher count of travels (24.89%)

Figure 18(b) and 18(c) show the helping behavior of the object over time. Given that all objects start approximately at the same time and that they start



(a) MRR (b) Cnt-Friendship-Requested (c) QoF

Fig. 19: Attraction

off with no friends at all, objects with strong conscientiousness will help new objects to establish themselves (make new friends) in the network. Figure 18(a) shows, that Helpfulness objects tend to have a bigger neighborhood at the beginning because they decide to go into a friendship to help others if they have fewer friends. Figure 18(b) shows that the Overall-Similarity in the beginning, when they to be helpful, is low.

Attraction.

Objects which are strong in agreeableness are selected more often as friends compared to others. To verify this behavior we observe how these objects get selected from others. The dataset covers 165 attraction objects.

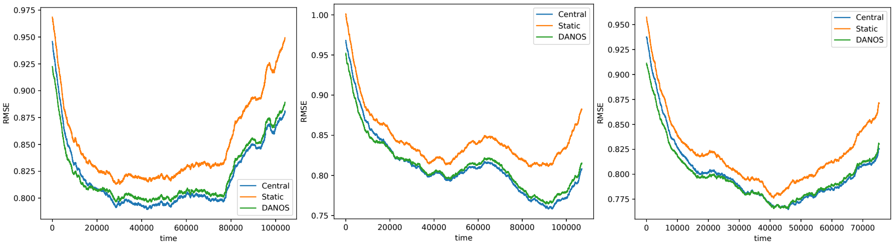
The results in Table 14 and Figure 19 show that the MRR increases by 1.71% with a larger Neighbourhood (8.52%) and a decrease in Overall-Similarity (-13.56%) and QoF (-11.59%). Preference-Diversity and Profile-Similarities change less (<1%). From the network's perspective, the number of crawls increases (61.35%) and the Num-Friendship-Requests increases by 32-28%. These observations show that attraction objects get selected as friends more times compared to the case of no HCB.

5.4 RQ$_4$: Effectiveness

In the last round of experiments, we compare the effectiveness of the DANOS recommender against two baselines, i.e., Central and Static. We expect that on the one hand DANOS would be able to generate as effective recommendations compared to Central (H_5), and on the other hand DANOS would yield considerably more advantages in comparison to Static (H_6).

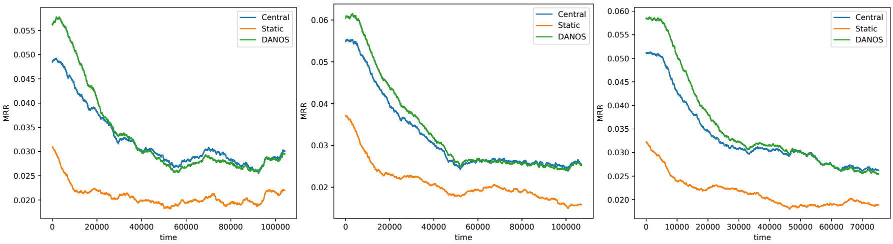
Dataset Samples.

First, we consider three different dataset samples and compute recommender effectiveness metrics. In this round, we keep the number of cells to 1. Results are summarized in Table 15, where we see similar effectiveness across samples.



(a) Sample 1 (b) Sample 2 (c) Sample 3

Fig. 20: Recommender effectiveness (RMSE) in different dataset samples



(a) Sample 1 (b) Sample 2 (c) Sample 3

Fig. 21: Recommender effectiveness (MRR) in different dataset samples

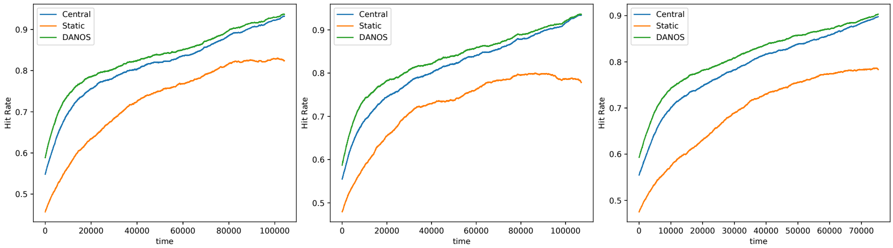
It is important to note that DANOS achieves comparable or even better effectiveness compared to Central. This occurs despite Central having access to the entire information. In contrast Static has considerably worse performance.

To understand the reason for these observations, we need to examine in detail how effectiveness evolves over time. Figure 20 show RMSE over time for the three samples. What we observe is that at the beginning, DANOS has a better (lower) RMSE than Central in all samples. Over time however, Central improves and becomes better. This happens because at the beginning there is little rating/feedback information that Central can exploit. In contrast DANOS is able to rely on other sources of information, namely profile and preferences similarities, which help it create meaningful neighborhood.

The same effect appears in MRR and in a smaller scale in Hit Rate, as shown in Figures 21 and 22, respectively. Therefore, the explanation for why DANOS appears overall more effective than the central approach is the fact that it performs strongly in the beginning when it optimally uses the scarce information available.

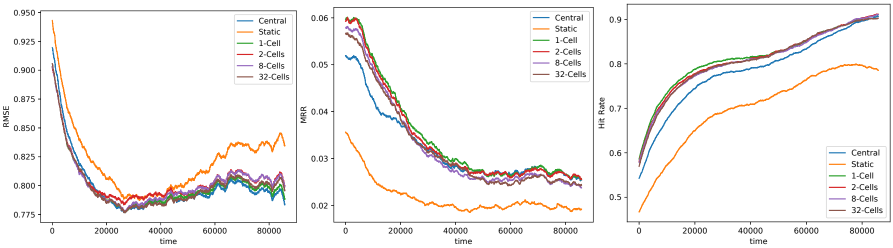
Global Decentralization Level.

In the last experiment, we compare DANOS at different global decentralization levels (increasing the number of cells from 1 up to 32) against the baselines.



(a) Sample 1 (b) Sample 2 (c) Sample 3

Fig. 22: Recommender effectiveness (Hit Rate) in different dataset samples



(a) RMSE (b) MRR (c) Hit Rate

Fig. 23: Recommender effectiveness at different global decentralization levels

Table 16 summarizes the evaluation metrics. It comes as no surprise that decentralization comes at the expense of recommender effectiveness; the best setting for DANOS is with one cell. Figure 23 validates this observation by presenting the evolution of all metrics over time. Interestingly, the static approach is less effective than even the highest decentralization tested. This further supports the idea that dynamically and intelligently adapting the neighborhoods gives an effectiveness level that rivals that of a centralized approach.

5.5 Evaluation Interpretation and Lessons Learned

In this section, we revisit the research questions and the tested hypotheses and generate key take-aways. In brief, the evaluation has validated the three design principles of DANOS, and thus demonstrates that they can successfully address the current SIoT challenges.

Dynamicity.

The definition of task-dependent quality measures for a social neighborhood enables objects to act autonomously, via an online learning process, so as to continuously improve the utility of their social connections. Figures 7 and 9 show that convergence is achieved quickly in almost all cases, albeit at different convergence rates with respect to hyperparameter values. We therefore accept

H_1. Moreover, Figures 8 and 10, show that slow convergence (e.g., when the learning rate is 0.01 or the learning period is 30) leads to reduced recommendation effectiveness. This implies that we can tune the online learning process for quick convergence, and at the same time enjoy an increase in effectiveness.

Moreover, we find that dynamicity results in higher effectiveness for the recommendation task compared to a static approach. Figures 20–22 clearly demonstrate the superiority of our dynamic approach compared to Static, where objects do not adapt and evolve their connections over time. Thus we accept H_6 for all effectiveness metrics.

Decentralization.

Decentralization in DANOS is achieved globally by the two-level virtual space, and locally by the friendship neighborhood of each object. Local decentralization can be achieved with little impact on task effectiveness. Specifically, Table 5 depicts that increasing the local decentralization level leads to more effective recommendations. We further observe that the rate of effectiveness improvement decreases as we increase neighborhood size. This observation is more apparent for RMSE and MRR, and less so for Hit-Rate and List-Size. Therefore, we accept H_2 in terms of RMSE and MRR.

Global decentralization via the two-level virtual space (areas and cells), where objects travel to meet others, leads to cell specialization over time (via object-object feedback) that enables the efficient and effective discovery of relevant information for a task. Table 6 shows cell specialization, in terms of specific notions, at different global decentralization levels. Therefore, we accept H_3 when specialization is captured in terms of mean distance of rating and preferences.

Moreover, decentralization and the implementation of access control for the flow of information among objects enable *privacy by design*. More importantly, privacy comes at almost no cost to effectiveness. Figures 20–22 demonstrate that the DANOS recommender achieves almost as, and in some cases more, effective recommendations than a no-privacy centralized approach, leading us to accept H_5 for all effectiveness metrics.

Anthropomorphism.

DANOS advocates the enrichment of SIoT objects' profiles with human-like traits, which translate to specific object behavioral patterns, so as to subjectively influence the affinity determined from the perspective of targeted objects towards their peers. Our evaluation revealed that using this approach we can influence the establishment of relationships or connections between pairs of objects that are tested for compatibility, and can also influence how and how often object pairs are selected influencing the formation and utilization of the network. Through incorporation of principal and behavioral attributes, an object can better represent the interests of its owner, and can form connections with those partner objects best suited to assist in the performance of its functions. Table 9 shows that human-centered behavior leads to more effective recommendations in terms of MRR. Moreover, Table 6 shows that HCB has a positive impact on cell specialization, as all distance metrics decrease. As a result, we conclude that H_4 is accepted.

5.6 Validity and Limitations

The acceptance of DANOS experimentation method, execution and outcomes primarily influenced by overarching evaluation principles that cover its internal and external validity. The former refers to the accuracy of the dataset used during the evaluation process and in accordance the conclusions and interpretations drawn upon the data, whilst the latter to which extent those conclusions can be generalized and replicated also with respect to other application contexts and implementation fields [99].

Aiming at increasing the internal validity of our evaluation, we carefully selected a dataset that would be suitable (i.e., number of attributes, types and association of variables) to our research objectives and the various experimentation objectives, i.e., a structure easy to consume and a set of properties that would comply with the theoretical perspectives and computational methods of the proposed architecture. Hence, after an extensive research we decided to qualify MovieLens 1M dataset, due to the lack, to our knowledge, of a dataset in the area of SIoT that would have been a good fit to our purposes. Main selection requirements emphasized upon data availability (or possibility for association with prior research outcomes and theoretical testimonies for extracting valid inferences of data values for attributes that were not available, i.e., enriched the dataset with users' personality based on the correlation of movies genres and personality traits) for evaluation of user data (e.g., demographics, items, ratings, preferences, personality); availability of large amounts of user feedback; human-centered interaction behavior and similarity metrics of objects; recommendation effectiveness using DANOS compared to other baseline settings; DANOS decentralized and dynamic setting performance in relation to static and centralized settings.

Regarding external validity, given that future experimentation settings and studies will contribute to the external validity of the reported research, we expect that using the DANOS decentralized architecture and recommendation engine in different SIoT contexts and contents (probably supported with the availability of more direct datasets) could improve the overall quality of information exchange, recommendations effectiveness and users' experience regarding products and services. The latter argument can be supported by the outcomes of the current evaluation phases whereby we can observe that similarity weights towards user profiling and ratings fed and dynamically maintained over time by users' feedback overrule the more static initial perceptions triggered by more generic types of information, e.g., genre preferences. Furthermore, there is a clear tendency towards a more optimized utilization of network resources due to the human-centred behavior of objects and cells' specialization over time.

Nonetheless, this work has some limitations which primarily refer to the use of MovieLens dataset instead of one that would be more directly connected to the SIoT domain. As mentioned above, this was a thoughtful decision following a thorough investigation in the area, led by our current priority to evaluate the theoretical dimensions and technical viability of the proposed solution as well as the added value which is bound to the use of innovative procedures and elements. Further limitations include: (a) The use and extraction of the personality traits. As the personality characteristics of users were not available in the employed dataset we used prior literature's findings (see Section 5.1) to extract respective associations between users' preference to movie genres and personality types. This way, we managed to integrate into our dataset a well received theoretical dimension that has led us to unique and innovative findings during its application in the objects' profiles and their HCB during operation. However, considering that there is a lack of a uniform methodology for applying triangulation, the product of such analysis could be regarded as relative (e.g., acknowledge other interpretations of data associations with equal importance as the ones considered in this work), as its validity it is rather improbable to be assessed in terms of an accuracy measure stemmed from a ground truth; and (b) the adoption of the HCB by the objects based on the various personality types adhered to specific interpretations of the behaviors that are generated from the combination of the individual personality traits. Such behaviors are the outcome of an offline analysis based on previous research findings with respect to the impact that Big Five personality traits have on interactions between entities in the human social networks (see Section 4.2.9). Consequently, given the inherent fuzziness that these individual characteristics entail, the same combinations in the range of each individual characteristic might produce different behaviors that might drive a different computational process and design of experimentation conditions. In addition, we have to recognize that although the specific behaviors simulate adequately the various interactions, decisions and establishment of friendships between the objects in our experimentation setting, we should expect that it would be particularly challenging for objects to have as many external friends in a real-life SIoT scenario. Eventually, an important limitation is that the reported evaluation does not include interactions with real end-users, in their physical environment and daily routine so to assess also the ecological validity [100] of this work. This would allow us to evaluate the actual impact and accuracy of the proposed theoretical model, architecture, and effectiveness of recommendations on a user group through their real-time engagement and interaction with DANOS.

6 Conclusion

Inevitably the creation of intelligent methods and processes in the Area of SIoT can facilitate the more harmonious co-existence of humans and things. It is now acknowledged that smart objects can play a crucial role in the everyday life of users, supporting their main tasks, handling subsequent ones, or finding services and information that satisfy specific needs and intents. Although, the gap between the objects and their owners is being progressively bridged, compensating on each others weaknesses, there is still a huge potential that could be exploited with the use of users' intrinsic characteristics, as those dictated by the Theory of Individual Differences, and has been proved successful elsewhere, like in the HSN. Related research work has shown that human factors, like for example personality traits, have direct influence on the creation, modelling and explanation of social interactions that in turn might produce beneficial results for the users.

In this work we explore the possibility of utilizing the research outcomes and lessons learned from the field of HSN for proposing a human-centered decentralized SIoT architecture, that enables the intelligent travelling (search and finding process stages), interaction and establishment of friendships between autonomous smart objects. Accordingly, recommendations are provided on given users' requests and intents, minimizing at the same time the network complexity and load. More specifically, the proposed recommendation engine in SIoT runs over an agent-based decentralized architecture, with the aim to create a good pairing between the objects for delivering best fit recommendations to their owners. Main innovation points of the current solution are: the execution is in a fully dynamic social network; there is no full view of the network and historical information; the information distributed among agents is not centrally controlled and respects security and privacy concerns; the network formation is optimized over time leading in better recommendation effectiveness than a centralized approach, as evidenced by experiments in a large-scale dataset. Furthermore, we developed a system (and simulator), DANOS, that is composed of 3 overarching parts: The *User* and *Object*, the *SIoT Manager*, and the *SIoT Space* components. We simulate a real-life scenario, detailing the workflow process of an object discovering and recommending optimized sets of movies alternatives to its owner after it has interacted and exchanged information with its friends in the network of objects.

We performed an extensive experimental evaluation in 4 distinctive phases, finding the DANOS recommender can be as effective as a centralized approach that has complete knowledge of the entire data. This is thanks to the learning process that each object run, and the specialization effect of DANOS cells, that result in object creating meaningful neighborhoods over time. Moreover, we find that the human-centered behavior is adopted by objects as expected, and it can also bring benefits in recommendation effectiveness. Accordingly, the highlighted lessons learned from the proposed approach are that the three pillars of DANOS, *dynamicity*, *decentralization*, and *anthropomorphism*, address the current *SIoT challenges* and more importantly lead to increased *effectiveness* for data-hungry IoT tasks, such as recommendations. We thus advocate the design of SIoT frameworks that follow these three design principles.

For the future work our main concern is to verify the applicability and usefulness of DANOS not only on a network but also on a user level, we will follow-up with the design and development of the owner-object interface and the cross-validation of the platform's usability and impact from a user's perspective. In addition, in our immediate plans is to design an experimentation setting applied in an actual SIoT environment, engaging real-life objects and related datasets, so to verify the effectiveness of the current simulation and innovation of the proposed theoretical models, architecture and algorithms. This way, we expect to offer a more holistic benefit and added-value to the current SIoT research and architectures.

References

[1] Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer networks **54**(15), 2787–2805 (2010)

[2] Greengard, S.: The Internet of Things. MIT press, Cambridge, MA (2015)

[3] Research, Markets: LTE IoT Market by Technology, Service, Industry, and Region - Global Forecast to 2023. [Online; accessed 07 January 2020] https://www.researchandmarkets.com/publication/m6e7ijf/4753838 (2019)

[4] Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. Computer networks **56**(16), 3594–3608 (2012)

[5] Cheng, C., Zhang, C., Qiu, X., Ji, Y.: The social web of things (swot)-structuring an integrated social network for human, things and services. JCP **9**(2), 345–352 (2014)

[6] Atzori, L., Iera, A., Morabito, G.: Siot: Giving a social structure to the internet of things. IEEE communications letters **15**(11), 1193–1195 (2011)

[7] Asl, H.Z., Iera, A., Atzori, L., Morabito, G.: How often social objects meet each other? analysis of the properties of a social network of iot devices based on real data. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 2804–2809 (2013). IEEE

[8] Nitti, M., Girau, R., Atzori, L., Iera, A., Morabito, G.: A subjective model for trustworthiness evaluation in the social internet of things. In: 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio communications-(PIMRC), pp. 18–23 (2012). IEEE

[9] Nitti, M., Atzori, L., Cvijikj, I.P.: Network navigability in the social internet of things. In: 2014 IEEE World Forum on Internet of Things

(WF-IoT), pp. 405–410 (2014). IEEE

[10] Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender Systems Handbook, pp. 1–35. Springer, Boston, MA (2011)

[11] Felfernig, A., Erdeniz, S.P., Jeran, M., Akcay, A., Azzoni, P., Maiero, M., Doukas, C.: Recommendation technologies for iot edge devices. Procedia Computer Science **110**, 504–509 (2017)

[12] Felfernig, A., Boratto, L., Stettinger, M., Tkalčič, M.: Group Recommender Systems: An Introduction. Springer, Cham (2018)

[13] Masthoff, J.: Group recommender systems: Combining individual models. In: Recommender Systems Handbook, pp. 677–702. Springer, Boston, MA (2011)

[14] Burke, R.: Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction **12**(4), 331–370 (2002)

[15] Felfernig, A., Erdeniz, S.P., Uran, C., Reiterer, S., Atas, M., Tran, T.N.T., Azzoni, P., Király, C., Dolui, K.: An overview of recommender systems in the internet of things. J. Intell. Inf. Syst. **52**(2), 285–309 (2019). https://doi.org/10.1007/s10844-018-0530-7

[16] Saleem, Y., Crespi, N., Rehmani, M.H., Copeland, R., Hussein, D., Bertin, E.: Exploitation of social iot for recommendation services. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 359–364 (2016). IEEE

[17] Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (SIoT) - When social networks meet the internet of things: Concept, architecture and network characterization. Computer Networks **56**(16), 3594–3608 (2012). https://doi.org/10.1016/j.comnet.2012.07.010

[18] Goldberg, L.R.: An alternative description of personality: the big-five factor structure. psycnet.apa.org (1990)

[19] Mashal, I., Alsaryrah, O., Chung, T.-Y.: Performance evaluation of recommendation algorithms on internet of things services. Physica A: Statistical Mechanics and its Applications **451**, 646–656 (2016)

[20] Lee, J.-S., Ko, I.-Y.: Service recommendation for user groups in internet of things environments using member organization-based group similarity measures. In: 2016 IEEE International Conference on Web Services (ICWS), pp. 276–283 (2016). IEEE

[21] Magerkurth, C., Sperner, K., Meyer, S., Strohbach, M.: Towards context-aware retail environments: An infrastructure perspective. Mobile Interaction in Retail Environments (MIRE 2011), Stockholm, Sweden (2011)

[22] Valtolina, S., Mesiti, M., Barricelli, B.: User-centered recommendation services in internet of things era. In: CoPDA2014 Workshop. Como, Italy (2014)

[23] Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM, pp. 931–940 (2008). https://doi.org/10.1145/1458082.1458205. http://doi.acm.org/10.1145/1458082.1458205

[24] Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011). https://doi.org/10.1145/1935826.1935877. http://doi.acm.org/10.1145/1935826.1935877

[25] Lee, J., Kim, S., Lebanon, G., Singer, Y., Bengio, S.: LLORMA: local low-rank matrix approximation. Journal of Machine Learning Research **17**, 15–11524 (2016)

[26] Baraglia, R., Dazzi, P., Mordacchini, M., Ricci, L.: A peer-to-peer recommender system for self-emerging user communities based on gossip overlays. Journal of Computer and System Sciences **79**(2), 291–308 (2013)

[27] Wang, Z., Liu, X., Chang, S., Zhou, J., Qi, G.-J., Huang, T.S.: Decentralized recommender systems. arXiv preprint arXiv:1503.01647 (2015)

[28] Barbosa, L.N., Gemmell, J., Horvath, M., Heimfarth, T.: Distributed user-based collaborative filtering on an opportunistic network. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 266–273 (2018). IEEE

[29] Cena, F., Console, L., Matassa, A., Torre, I.: Multi-dimensional intelligence in smart physical objects. Information Systems Frontiers **21**(2), 383–404 (2019)

[30] Rad, M.M., Rahmani, A.M., Sahafi, A., Qader, N.N.: Social internet of things: vision, challenges, and trends. Human-centric Computing and Information Sciences **10**(1), 1–40 (2020)

[31] Defiebre, D., Germanakos, P.: A human-centred business scenario in siot– the case of danos framework. In: IFIP Conference on Human-Computer Interaction, pp. 579–583 (2019). Springer

[32] Defiebre, D., Germanakos, P.: Towards a human-centered model in siot - enhancing the interaction behaviour of things with personality traits. In: Proc. of the 5th IEEE International Conference on Internet of People (2019). IEEE

[33] Defiebre, D., Germanakos, P., Sacharidis, D.: Danos: A human-centered decentralized simulator in siot. In: UMAP Adjunct (2020)

[34] Defiebre, D., Sacharidis, D., Germanakos, P.: A decentralized recommendation engine in the social internet of things. In: UMAP Adjunct (2020)

[35] Ashton, K., *et al.*: That 'internet of things' thing. RFID journal **22**(7), 97–114 (2009)

[36] Ursino, D., Virgili, L.: Humanizing iot: Defining the profile and the reliability of a thing in a multi-iot scenario. In: Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications, pp. 51–76. Springer, Cham (2020)

[37] Panayiotou, C., Samaras, G.: mpersona: personalized portals for the wireless user: An agent approach. Mobile Networks and Applications **9**(6), 663–677 (2004)

[38] Delicato, F.C., Pirmez, L., da Costa Carmo, L.F.R.: Fenix–personalized information filtering system for www pages. Internet Research (2001)

[39] Roopa, M., Pattar, S., Buyya, R., Venugopal, K.R., Iyengar, S., Patnaik, L.: Social internet of things (siot): Foundations, thrust areas, systematic review and future directions. Computer Communications **139**, 32–57 (2019)

[40] Cheng, C., Zhang, C., Qiu, X., Ji, Y.: The Social Web of Things (SWoT)-Structuring an Integrated Social Network for Human, Things and Services. Journal of Computers **9**(2), 345–352 (2014). https://doi.org/10.4304/jcp.9.2.345-352

[41] Lianhong Ding, Peng Shi, Liu, B.: The clustering of Internet, Internet of Things and social network. In: Third International Symposium on Knowledge Acquisition and Modeling, pp. 417–420. IEEE, New York (2010). https://doi.org/10.1109/KAM.2010.5646274. http://ieeexplore.ieee.org/document/5646274/

[42] Kim, J.E., Maron, A., Mosse, D.: Socialite: A flexible framework for social internet of things. In: 16th IEEE International Conference on Mobile Data Management, vol. 1, pp. 94–103 (2015). IEEE

[43] Kranz, M., Roalter, L., Michahelles, F.: Things that twitter: social networks and the internet of things. In What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010). (2010)

[44] Guinard, D., Fischer, M., Trifa, V.: Sharing using social networks in a composable web of things: Pervasive computing and communications workshops. 8th IEEE International Conference, 702–707 (2010). https://doi.org/10.1109/PERCOMW.2010.5470524

[45] Abdul, R., Paul, A., Gul, M., Hong, W.-H., Seo, H., *et al.*: Exploiting small world problems in a siot environment. Energies **11**(8), 2089 (2018)

[46] Kasnesis, P., Toumanidis, L., Kogias, D., Patrikakis, C.Z., Venieris, I.S.: Assist: An agent-based siot simulator. In: IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 353–358 (2016). IEEE

[47] Pintus, A., Carboni, D., Serra, A., Manchinu, A.: Humanizing the Internet of Things. In: 11th International Conference on Web Information Systems and Technologies (WEBIST2015) (2015). https://doi.org/10.5220/0005475704980503

[48] Koreshoff, T.L., Leong, T.W., Robertson, T.: Approaching a human-centred internet of things. In: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, pp. 363–366 (2013). ACM

[49] Fiske, A.P.: The four elementary forms of sociality: Framework for a unified theory of social relations. Psychological Review (1992). https://doi.org/10.1037/0033-295X.99.4.689

[50] Jung, J., Chun, S., Jin, X., Lee, K.-H.: Quantitative computation of social strength in social internet of things. IEEE Internet of Things Journal **5**(5), 4066–4075 (2018)

[51] Roopa, M., Valla, D., Buyya, R., Venugopal, K., Iyengar, S., Patnaik, L.: Sssss: Search for social similar smart objects in siot. In: 2018 Fourteenth International Conference on Information Processing (ICINPRO), pp. 1–6 (2018). IEEE

[52] Stern, W.: Über Psychologie der Individuellen Differenzen: Ideen zu Einer Differentiellen Psychologie" vol. 12. JA Barth, Leipzig (1900)

[53] Deary, I.J.: Human intelligence differences: a recent history. Trends in cognitive sciences **5**(3), 127–130 (2001)

[54] Picard, R.W.: Affective Computing. MIT press, Cambridge, MA (2000)

[55] Germanakos, P., Tsianos, N., Lekkas, Z., Mourlas, C., Samaras, G.: Capturing essential intrinsic user behaviour values for the design of comprehensive web-based personalized environments. Computers in Human Behavior **24**(4), 1434–1451 (2008)

[56] McCrae, R.R., John, O.P.: An introduction to the five-factor model and its applications. Journal of personality **60**(2), 175–215 (1992)

[57] Soldz, S., Vaillant, G.E.: The big five personality traits and the life course: A 45-year longitudinal study. Journal of Research in Personality **33**(2), 208–232 (1999)

[58] Burger, J.M.: Desire for Control: Personality, Social and Clinical Perspectives. Springer, Boston, MA (2013)

[59] Kleanthous, S., Herodotou, C., Samaras, G., Germanakos, P.: Detecting personality traces in users' social activity. In: International Conference on Social Computing and Social Media, pp. 287–297 (2016). Springer

[60] Van Lankveld, G., Spronck, P., Van den Herik, J., Arntz, A.: Games as personality profiling tools. In: 2011 IEEE Conference on Computational Intelligence and Games (CIG'11), pp. 197–202 (2011). IEEE

[61] Amichai-Hamburger, Y., Vinitzky, G.: Social network use and personality. Computers in Human Behavior **26**(6), 1289–1295 (2010). https://doi.org/10.1016/j.chb.2010.03.018

[62] Ross, C., Orr, E.S., Sisic, M., Arseneault, J.M., Simmering, M.G., Orr, R.R.: Personality and motivations associated with Facebook use. Computers in Human Behavior (2009). https://doi.org/10.1016/j.chb.2008.12.024

[63] Ferwerda, B., Tkalcic, M.: You are what you post: What the content of instagram pictures tells about users' personality. CEUR Workshop Proceedings. CEUR-WS.org, Aachen (2018)

[64] Quercia, D., Kosinski, M., Stillwell, D., Crowcroft, J.: Our twitter profiles, our selves: Predicting personality with twitter. In: Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/-SocialCom 2011 (2011). https://doi.org/10.1109/PASSAT/SocialCom.2011.26

[65] Hamburger, Y.A., Ben-Artzi, E.: The relationship between extraversion and neuroticism and the different uses of the internet. Computers in human behavior **16**(4), 441–449 (2000)

[66] M., S., W., B., S., B., J., D., van M., A., Selfhout, M., Burk, W., Branje, S., Denissen, J., van Aken, M., Meeus, W.: Emerging late adolescent friendship networks and big five personality traits: A social network approach. Journal of personality (2010). https://doi.org/10.1111/j.1467-6494.2010.00625.x

[67] McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual review of sociology **27**(1), 415–444 (2001)

[68] Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S., Seada, K.: Enhancing group recommendation by incorporating social relationship interactions. In: GROUP, pp. 97–106. ACM, New York, NY, USA (2010). https://doi.org/10.1145/1880071.1880087. https://doi.org/10.1145/1880071.1880087

[69] Lewis, K., Gonzalez, M., Kaufman, J.: Social selection and peer influence in an online social network. Proceedings of the National Academy of Sciences **109**(1), 68–72 (2012)

[70] Sánchez, L.Q., Recio-García, J.A., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. ACM TIST **4**(1), 8–1830 (2013). https://doi.org/10.1145/2414425.2414433

[71] Massa, P., Avesani, P.: Trust-aware collaborative filtering for recommender systems. In: CoopIS, pp. 492–508 (2004). https://doi.org/10.1007/978-3-540-30468-5_31. http://dx.doi.org/10.1007/978-3-540-30468-5_31

[72] Massa, P., Avesani, P.: Trust-aware recommender systems. In: RecSys, pp. 17–24 (2007). https://doi.org/10.1145/1297231.1297235. http://doi.acm.org/10.1145/1297231.1297235

[73] Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: ACM SIGIR, pp. 203–210 (2009). https://doi.org/10.1145/1571941.1571978. http://doi.acm.org/10.1145/1571941.1571978

[74] Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: ACM RecSys, pp. 135–142 (2010). https://doi.org/10.1145/1864708.1864736. http://doi.acm.org/10.1145/1864708.1864736

[75] Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A., Radha, H.: Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In: ACM RecSys, pp. 51–58 (2015). https://doi.org/10.1145/2792838.2800198. http://doi.acm.org/10.1145/2792838.2800198

[76] Li, H., Wu, D., Tang, W., Mamoulis, N.: Overlapping community regularization for rating prediction in social recommender systems. In: ACM RecSys, pp. 27–34 (2015). https://doi.org/10.1145/2792838.2800171. http://doi.acm.org/10.1145/2792838.2800171

[77] Rafailidis, D., Crestani, F.: Learning to rank with trust and distrust in recommender systems. In: ACM RecSys, pp. 5–13. ACM, New York, NY, USA (2017)

[78] Zhao, H., Yao, Q., Kwok, J.T., Lee, D.L.: Collaborative filtering with social local models. In: ICDM, pp. 645–654. IEEE Computer Society, New York (2017). https://doi.org/10.1109/ICDM.2017.74. https://doi.org/10.1109/ICDM.2017.74

[79] Zhang, Z., Liu, H.: Social recommendation model combining trust propagation and sequential behaviors. Applied Intelligence **43**(3), 695–706 (2015)

[80] Ziegler, C.-N.: Towards decentralized recommender systems. PhD thesis, Albert-Ludwigs-Universität Freiburg (2005)

[81] Kermarrec, A.-M., Leroy, V., Moin, A., Thraves, C.: Application of random walks to decentralized recommender systems. In: International Conference On Principles Of Distributed Systems, pp. 48–63 (2010). Springer

[82] Han, P., Xie, B., Yang, F., Shen, R.: A scalable p2p recommender system based on distributed collaborative filtering. Expert systems with applications **27**(2), 203–210 (2004)

[83] Yang, W.-S., Hwang, S.-Y.: itravel: A recommender system in mobile peer-to-peer environment. Journal of Systems and Software **86**(1), 12–20 (2013)

[84] Beierle, F., Eichinger, T.: Collaborating with users in proximity for decentralized mobile recommender systems. arXiv preprint arXiv:1906.03114 (2019)

[85] Munoz-Organero, M., Ramírez-González, G.A., Munoz-Merino, P.J., Kloos, C.D.: A collaborative recommender system based on space-time similarities. IEEE Pervasive Computing **9**(3), 81–87 (2010)

[86] Benouaret, I., Lenne, D.: Personalizing the museum experience through context-aware recommendations. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 743–748 (2015). IEEE

[87] Ray, P.P.: Generic internet of things architecture for smart sports. In:

2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 405–410 (2015). IEEE

[88] Leitner, G., Felfernig, A., Fercher, A.J., Hitz, M.: Disseminating ambient assisted living in rural areas. Sensors **14**(8), 13496–13531 (2014)

[89] Mohammadi, V., Rahmani, A.M., Darwesh, A.M., Sahafi, A.: Trust-based recommendation systems in internet of things: a systematic literature review. Human-centric Computing and Information Sciences **9**(1), 21 (2019)

[90] Amato, F., Mazzeo, A., Moscato, V., Picariello, A.: A recommendation system for browsing of multimedia collections in the internet of things. In: Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence, pp. 391–411. Springer, Berlin Heidelberg (2013)

[91] Forestiero, A.: Multi-agent recommendation system in internet of things. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), pp. 772–775 (2017). IEEE

[92] Chen, Z., Ling, R., Huang, C., Zhu, X.: A scheme of access service recommendation for the social internet of things. Int. J. Communication Systems **29**(4), 694–706 (2016). https://doi.org/10.1002/dac.2930

[93] Lin, Z., Dong, L.: Clarifying trust in social internet of things. IEEE Trans. Knowl. Data Eng. **30**(2), 234–248 (2018). https://doi.org/10.1109/TKDE.2017.2762678

[94] Gulati, N., Kaur, P.D.: When things become friends: a semantic perspective on the social internet of things. In: Smart Innovations in Communication and Computational Sciences, pp. 149–159. Springer, Singapore (2019)

[95] Goldberg, L.R.: An alternative "description of personality": the big-five factor structure. Journal of personality and social psychology **59**(6), 1216 (1990)

[96] Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR, pp. 335–336. ACM, New York, NY, USA (1998). https://doi.org/10.1145/290941.291025. https://doi.org/10.1145/290941.291025

[97] Santos, R.L.T., Macdonald, C., Ounis, I.: Exploiting query reformulations for web search result diversification. In: Rappa, M., Jones, P., Freire, J., Chakrabarti, S. (eds.) WWW, pp. 881–890. ACM,

New York, NY, USA (2010). https://doi.org/10.1145/1772690.1772780. https://doi.org/10.1145/1772690.1772780

[98] Cantador, I., Fernández-Tobías, I., Bellogín, A.: Relating personality types with user preferences in multiple entertainment domains. CEUR Workshop Proceedings, vol. 997. CEUR-WS.org, Aachen (2013). http://ceur-ws.org/Vol-997/empire2013_paper_2.pdf

[99] Cook, T.D., Campbell, D.T., Day, A.: Quasi-experimentation: Design & Analysis Issues for Field Settings vol. 351. Houghton Mifflin, Boston (1979)

[100] Brewer, M.B., Crano, W.D.: Research design and issues of validity. Handbook of research methods in social and personality psychology, 3–16 (2000)

Daniel Defiebre, M.Sc., is CTO and Co-Founder of the Startup NextX. NextX provides software solutions for employer branding, career orientation, and education with a focus on Data Privacy and Security. His main task is to drive and manage software development, develop human-centered architectures, and create algorithms for matchmaking. His educational background is interdisciplinary in the cross-borders of Computer Science and Psychology. Before his current task, he worked as a researcher in Human-Centered Architecture and Algorithms at SAP and worked as a consultant in software development projects for the German government. Additionally to his main job he has a consultant praxis for career and job development to support people with their problems. He has published 4 publications in journals and conferences.

Dimitris Sacharidis, Ph.D., is an assistant professor at the CoDE Department of the Université Libre de Bruxelles. Prior to that he was an assistant professor at the Technical University of Vienna, and a Marie Skłodowska Curie fellow at the "Athena" Research Center and at the Hong Kong University of Science and Technology. He finished his PhD and undergraduate studies on Computer Engineering at the National Technical University of Athens, while in between he obtained an M.Sc. in Computer Science from the University of Southern California. His research interests include data science, data engineering, and recommender systems. He has served as a PC member and has been involved in the organization of top related conferences, and has acted as a reviewer and editorial member of associated journals.

Panagiotis Germanakos, Ph.D., is User Experience Research Expert - Instructor at SAP SE, leading the user research of product teams for delivering usable, high quality, human-centered solutions. He is also Research Scientist, serving as liaison between the business and academia, conducting basic/applied research, consulting and transferring knowledge that shapes innovations.

His educational background is interdisciplinary in the cross-borders of Computer Science, Psychology and Business Informatics disciplines, whereby his research focuses in UX, HCI, User Modeling, Adaptation & Personalization, and Human-centered Artificial Intelligence. He has over 120 publications in accredited scientific books, journals and conferences, incl. a co-authored monograph and his work has obtained a number of awards. He has also published 3 utility and 2 design patents, and he is co-founder/organizer of international scientific events and member of various editorial and program committees of top journals, conferences and workshops.

Table 1: DANOS Components and their Role

Component	**Description**	**Used For**
Object Profile	The data which describe the user, the object's experience in DANOS and his IoT Devices	Similarity Calculation between two objects and between the object and a cell proxy (virtual object). Includes the parameter for the HCB adaptation
User Adaptation	Processes user's requests, engines to build recommendations and adapt content to user's need	Build the local network of friendships, collect friends' ratings, runs the recommender and processing the feedback from the user. Takes information from the User Specifics to align content to the user
Cyberspace	Virtual location based framework to bring similar Objects together	Location based representation of intents and clusters to handle a huge amount of objects
SIOT Manager	Entering point for Object and Areas	Security, Administration, Right Management for Objects and Areas; Area Management.
Intent Manager	Forward the object to the area which handles it's intent	Compares the object's intent with the area's intents to forward the address of the area which handles the object's intent.
Area Manager	Registration Engine for Objects and Cells	Security, Administration, Right Management for Objects and Cells; Cell Management.
Schedule Manager	Generates a schedule for the objects, which cells the object should visit to find the most similar pairs	Compares the object profile with all cell proxies (like a synthetic object) to find the most similar cells.
Cell Manager	A cluster with functionality	Creates a virtual representation (Cell Proxy) from object profiles of the most successful interactions to define the cluster. The object Profile information will be feedbacked to the cell where a friendship was established if a recommendation was successfully to build the proxy.
Relationship Manager	Provides new friendships for a requested object	Calculates the object profile similarity from the requested object to all other objects in the cell. Adapt human-centered behavior to the calculated metric

Table 2: Interpretation of Personality Traits as Human-Centered Behavior

Personality Traits	**Definition**	**HCB Behaviour**
Extraversion (E) Extraverts are characterized by excitability, sociability, talkativeness, high amounts of emotional expressiveness	High in E like to start conversations, enjoy meeting people and sharing news	Strong in A → **Approaching**
Conscientiousness (C) High levels of thoughtfulness, with good impulse control and goal-directed behaviors	High in C relate to spending more time for preparation, organizing, attention to details, assisting co-workers for accomplishing tasks and solving problems	Strong in C → **Attraction**
Neuroticism (N) Neurotic people are characterized primarily by emotional instability	High in N tend to experience mood swings, anxiety, get upset easily and worry about many different things	High in N and Low in O → **Risk-Avoiding**
Openness to experience (O) People are characterized by imagination and insight	High in O tend to have a broad range of interests, to be more open to try new things, tackle new challenges and welcome thinking of new concepts	High in O and Low in N → **Risk-Taking**
Agreeableness (A) People are characterized by trust, altruism, kindness and affection	High in A are more cooperative, enjoy helping and contributing to the happiness of others	Strong in C → **Helpfullness**

Table 3: Recommender Effectiveness for Various Learning Rates for Fixed Learning Period of 10

Method	**RMSE**	**MRR**	**Hit-Rate**	**List-Size**
Learning-Rate-0.01	0.8030	0.0323	0.8097	834.9533
Learning-Rate-0.05	0.7998	**0.0345**	0.8151	**850.6137**
Learning-Rate-0.08	0.7992	0.0342	0.8168	846.5593
Learning-Rate-0.1	**0.7988**	0.0342	**0.8174**	847.0723
Learning-Rate-0.2	0.7989	0.0335	0.8171	846.8026

Table 4: Recommender Effectiveness for Various Learning Periods for Fixed Learning Rate of 0.05

Method	**RMSE**	**MRR**	**Hit-Rate**	**List-Size**
Learning-Period-5	0.8005	0.0342	0.8137	**854.7528**
Learning-Period-10	0.7998	**0.0345**	**0.8151**	850.6137
Learning-Period-15	**0.7985**	0.0343	0.8148	840.8003
Learning-Period-20	0.7992	0.0334	0.8141	838.7813
Learning-Period-30	0.7993	0.0332	0.8144	836.5000

Table 5: Recommender Effectiveness for Various Neighborhood Sizes

Method	RMSE	MRR	Hit-Rate	List-Size
Neighborhood-20	0.8010	0.0333	0.7510	675.7247
Neighborhood-30	0.7998	0.0345	0.8151	850.6137
Neighborhood-40	0.7980	0.0347	0.8535	987.8431
Neighborhood-50	**0.7973**	**0.0348**	**0.8785**	**1103.7953**

Table 6: Cell Specialization

Num. of Cells	Mean Profiles Dis.	Mean Ratings Dis.	Mean Pref. Dis.
4	0.006	0.459	0.124
8	0.008	0.455	0.116
32	0.011	0.432	0.114

Table 7: Number of Objects Per Personality Trait

Personality Trait	Num. of Objects Strong in	Num. of Objects Weak in
Openness	91	51
Conscientiousness	90	54
Extraversion	87	107
Agreeableness	165	41
Neuroticism	114	112

Table 8: Number of Objects Per Human-Centered Behavior

Approaching	Risk-Taking	Risk-Avoiding	Attraction	Helpfulness
87	7	17	165	90

Table 9: Overall Human-Centered Behavior Effect

Method	MRR	NB-Size	Overall-Similarity	QoF	Pref-Diversity	Pref-Similarity
No-HCB	0.0324	300	0.0317	0.0276	0.1506	0.8718
HCB	0.0331	307.828	0.0303	0.0265	0.1447	0.8786
Difference	2.16%	2.61%	-4.42%	-3.99%	-3.92%	0.78%

Method	Num-Schedule-Requests	Num-Friend-Requests	Num-Cell-Visits	Num-Travels
No-HCB	4.311.228	8.170.350	285.998	74.360.415
HCB	4.899.271	9.630.781	281.849	90.760.970
Difference	13.64%	17.87%	-1.45%	22.06%

Table 10: Cell Specialization with HCB

Num. of Cells	Mean Profiles Dis.	Mean Ratings Dis.	Mean Pref. Dis.
No-HCB	0.011	0.432	0.114
HCB	0.0038	0.4306	0.1134
Difference	-64.15%	-0.42%	-0.53%

Table 11: Considering Only Approaching Objects

Method	MRR	NB-Size	QoF	Pref-Diversity	Profile-Similarity	Rat-Similarity	Pref-Similarity
No-HCB	0.0324	300	0.028	0.114	1	0.535	0.904
HCB	0.0334	386.768	0.017	0.151	1	0.539	0.886
Difference	3.09%	28.92%	-39.71%	33.04%	0.00%	0.62%	-1.98%

Method	Num-Crawling	Num-F-Canceling	Visited-Cells	Num-Travels
No-HCB	3.137.322	26.238.786	257.987	59.975.143
HCB	10.868.426	36.172.749	288.032	79.657.671
Difference	246.42%	37.86%	11.65%	32.82%

Table 12: Considering Only Risk-Taking and Risk-Avoiding

Method	MRR	Overall-Similarity	QoF	Pref-Diversity	Profile-Similarity	Rat-Similarity	Pref-Similarity	Num-Crawling
Risk-Taking								
No-HCB	0.0518	0.0318	0.0273	0.2422	0.9996	0.531	0.6938	3.834.215
HCB	0.0473	0.0319	0.0267	0.2372	0.9998	0.5302	0.6948	3.800.847
Difference	-8.69%	0.31%	-2.20%	-2.06%	0.02%	-0.15%	0.14%	-0.87%
Risk-Avoiding								
No-HCB	0.0366	0.0318	0.028	0.1705	0.9998	0.5247	0.8494	2.068.909
HCB	0.0405	0.0257	0.0232	0.1735	0.9997	0.5222	0.8566	3.756.910
Difference	10.66%	-19.18%	-17.14%	1.76%	-0.01%	-0.48%	0.85%	81.59%

Table 13: Considering Only Helpfulness

Method	MRR	NB-Size	Overall-Similarity	QoF	Pref-Diversity	Pref-Similarity
No-HCB	0.0339	300000	0.032	0.028	0.159	0.8632
HCB	0.0371	311305	0.03	0.026	0.154	0.8726
Difference	9.44%	3.77%	-6.29%	-5.42%	-3.46%	1.09%

Method	Num-Crawling	Num-Friend-Requests	Num-Cell-Visits	Num-Travels
No-HCB	2.201.613	4.630.250	272.046	49.902.327
HCB	2.699.862	5.592.280	266.445	62.323.745
Difference	22.63%	20.78%	-2.06%	24.89%

Table 14: Considering Only Attraction

Method	MRR	NB-Size	Overall-Similarity	QoF	Pref-Diversity	Profile-Similarity
No-HCB	0.0292	300.000.000	0.032	0.028	0.142	1
HCB	0.0297	325.552.000	0.027	0.024	0.142	1
Difference	1.71%	8.52%	-13.56%	-11.59%	-0.21%	0.00%

Method	Rat-Similarity	Pref-Similarity	Count-Crawling	Count-F-Requested
No-HCB	0.537	0.882	3.798.444	7.061.257
HCB	0.535	0.888	6.128.851	9.340.685
Difference	-0.26%	0.68%	61.35%	32.28

Table 15: Recommendation Effectiveness in Different Dataset Samples

Method	**RMSE**	**MRR**	**Hit Rate**	**List Size**
Central (sample 1)	0.8200	0.0324	0.8141	959.6402
Static (sample 1)	0.8504	0.0208	0.7256	1014.2221
DANOS (sample 1)	0.8222	0.0334	0.8348	956.1681
Central (sample 2)	0.8111	0.0320	0.8137	983.8871
Static (sample 2)	0.8439	0.0209	0.7243	999.0893
DANOS (sample 2)	0.8124	0.0335	0.8349	987.6367
Central (sample 3)	0.8012	0.0332	0.7900	803.6469
Static (sample 3)	0.8210	0.0215	0.6920	861.5612
DANOS (sample 3)	0.7991	0.0353	0.8145	794.0707

Table 16: Recommendation Effectiveness at Various Global Decentralization Levels

Method	**RMSE**	**MRR**	**Hit Rate**	**List Size**
Central	0.7997	0.0322	0.7870	847.3950
Static	0.8215	0.0217	0.7009	909.3693
1 Cell	**0.7998**	**0.0345**	**0.8151**	850.6137
4 Cells	0.8046	0.0340	0.8112	862.8594
8 Cells	0.8028	0.0326	0.8084	856.0453
32 Cells	0.8001	0.0324	0.8083	**864.2234**