Crystal-Ball and Magic Wand Combined: Predicting Situations and Making Them Happen

Arkady Zaslavsky[1](✉), Ali Hassani[1], Pari Delir Haghighi[2], Antonio Robles-Kelly[1], and Panos K. Chrysanthis[3]

[1] Deakin University, Geelong, VIC 3217, Australia
arkady.zaslavsky@deakin.edu.au
[2] Monash University, Melbourne, VIC 3168, Australia
[3] University of Pittsburgh, Pittsburgh, PA 15260, USA

Abstract. The Internet of Things (IoT) envisions an ecosystem in which everyday objects are enhanced with sensing, computation, and communication capabilities. These 'smart' devices (i.e., IoT devices) can sense and collect considerable amounts of data and share it with each other via the Internet. This paper proposes an IoT middleware platform enhanced with context- and situation-prediction capability, called Context-Prediction-as-a-Service (CPaaS). CPaaS offers real-time context prediction capabilities to a variety of IoT applications as a service and enables more effective decision support using relevant validated dependable information. A number of use cases where CPaaS can be deployed are also discussed.

Keywords: Context · Context prediction · IoT · Situational awareness · Distributed context management platform

1 Introduction and Background

The Internet of Things (IoT) envisions an ecosystem in which everyday objects (e.g., refrigerator, air conditioner, smartphones, weather stations, cars, industrial robots, just to name a few) are enhanced with sensing, computation, and communication capabilities. These 'smart' devices (i.e., IoT devices) can sense and collect very large amounts of data and share it with each other via the Internet. Due to proliferation of IoT devices, their numbers are expected to reach 20 to 30 billion in 2021 [1]. It is then possible to build services that can share rich, useful and relevant information with users about an 'entity' and situation of interest. We define data external to such an entity and interpreted by the IoT application as context. Sharing context enables a wide range of context-aware and smart applications that can adapt their behavior according to the current context of one or several entities.

The need for contextual intelligence is a fundamental and critical factor for delivering IoT intelligence, efficiency, effectiveness, performance, and sustainability. Contextual intelligence enables intelligent interactions between IoT devices,

V. M. Vishnevskiy et al. (Eds.): DCCN 2020, LNCS 12563, pp. 683–697, 2020.
https://doi.org/10.1007/978-3-030-66471-8_52

such as sensors/actuators, mobile smart phones, smart vehicles to name a few. Context management platforms (CMP) for IoT applications are emerging as the ETSI (European Telecommunications Standards Institute) efforts on standardisation [2] prove. Existing CMPs only take the current context of IoT devices and entities into account. However, in many IoT applications, it is essential to predict the future context of IoT entities with acceptable confidence above specified threshold and to provide important relevant dependable real-time information and valuable recommendations for better decision support and actuation. For example, with context prediction and proactive adaptation, it would be possible to predict direction, speed and scope of fire spread to proactively deploy mobile hardware assets like robots and/or drones to set up virtual fences, herd, direct and save wildlife in case of bushfires, which are common to Australia.

To address this shortcoming of existing CMPs, in this paper, we propose a novel framework, coined Context-Prediction-as-a-Service (CPaaS). CPaaS can create new capabilities for context management platforms (CMPs) that enrich IoT applications with proactive and preventive behaviour. Such applications can predict the future context and complex situations and take pre-emptive actions, and continuously re-evaluate the impact of the actions and update/extend the existing knowledge. CPaaS will be extremely beneficial to building intelligent systems as it will provide the following novel components:

- A context prediction selector that can match the requirements of IoT applications to the prediction techniques and determine the most appropriate prediction technique.
- An evolutionary learning approach that constantly re-evaluates context prediction and updates the existing model with new knowledge.
- An actuation mechanism that enhances context prediction with the capability to support preventive and mitigating actions.
- A standard, formal and flexible context prediction model that will extend an existing context query language (CDQL) [3] developed by authors.

In the rest of this paper, we will first discuss the vision of CPaaS and highlight its importances and also the main challenges that need to be addressed to develop such a framework. Then, we will describe our proposed architecture for CPaaS and explain its underlying components.

2 Motivational Use-Case

An important application domain in dire need of context and situation prediction is wildlife conservation. Australia is home to distinctive wildlife and a number of extant species. While a great deal of effort is spent on wildlife conservation, bushfires pose a significant danger to already endangered species [4]. In 2009, Black Saturday bushfires in Victoria burnt over 450,000 hectares, killing about one million wild and domesticated animals, reported by RSPCA [5]. Recent bushfires in New South Wales, Victoria, Queensland, South Australia and Kangaroo Island destroyed millions of hectares, and left more than one billion animals dead [6].

In Kangaroo Island bushfires, it was estimated almost 30,000 koalas perished [6]. Figure 1 shows map of recent Australian bushfires during Summer of 2019–2020. Accurate prediction of fire behaviour, size and spread (including its shape, area and speed) can significantly help with mitigating and reducing the catastrophic effects of bushfires on wildlife and allow managing and sustaining fire-prone and safe ecosystems for them. A promising solution that can mitigate the effects of bushfires on wildlife is to create a virtual fence that translocates animals to safe fire-free areas. Virtual fence devices have been already tested in another significant threat to Australian wildlife, which is roadkill [7,8]. A trial of virtual fences in Tasmania over three years showed a reduction rate of 50% [7]. Virtual fencing is also used as an animal-friendly system to move or confine livestock. Context prediction can be used to predict future fire threats, and animals can be moved to a safe area (where no fire is predicted) by creating a virtual fence.

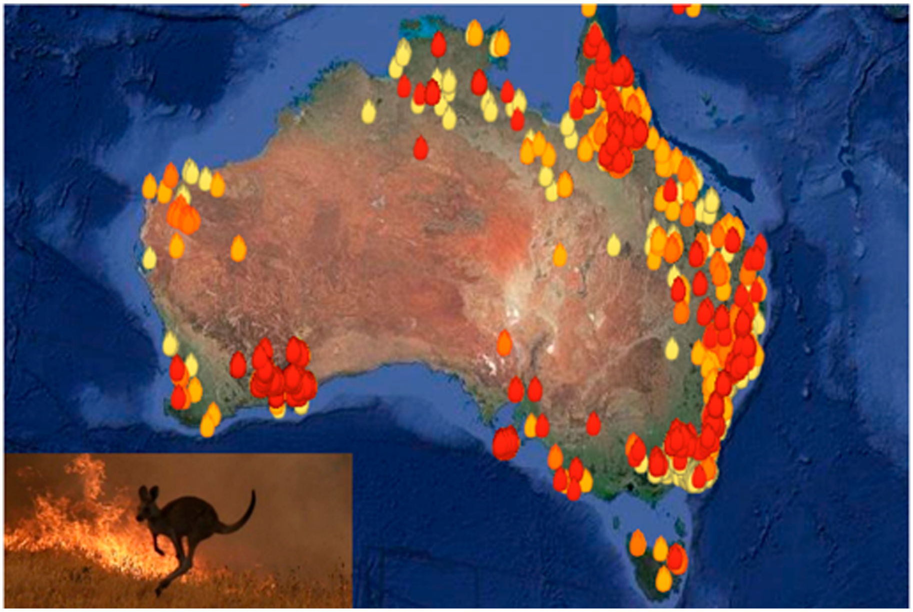

Fig. 1. Map of Australia with recent bishfires (https://sydneynews.sydney/sydney-news/1-billion-animals-perish-in-australian-bushfires/5762/, accessed on 12 September, 2020)

More practical and hotly needed motivational scenario is related to context-aware car parking availability prediction. Australia's capital cities have been transforming at a staggering pace. In 2011 the total population of the top five largest cities of Australia were around 13.5 million people [9]. Today, that figure is more than 16 million people, which means around 20% growth of population [10]. By 2055, the expected population of Australia's capital cities is predicted to reach more than 26 million people [11]. This growth in the population of the urban areas of Australia will undoubtedly put significant strain on the environment and infrastructures of the capital cities. For instance, without having a sophisticated plan to deal with the population explosion, in the near future, Australians will face several major problems such as traffic, air pollution, and water problems.

Therefore, to mitigate the possible negative impact of population growth, it is vital to design and develop effective and efficient solutions for better management of urban cities that contribute to environmental and urban sustainability and resilience.

One of the new raising challenges due to the population growth in large cities is searching for parking. As cities become more congested, the direct and indirect costs of parking are growing quickly. A survey completed in Melbourne and Sydney in 2014 showed that Australian drivers spend on average around 20 min a journey looking for parking during peak hour at busy areas of the city [12]. A similar study in the US has shown that Motorists spend an average of 17 h a year searching for carparks on-streets or in parking facilities [13]. Based on this report, the amount of wasted time, fuel, and emissions for each driver will add up to around $345 per year. This problem becomes worsen in large cities. For instance, In New York City drivers on average spend 107 h a year looking for parking spots, which about $2,243 in wasted time, fuel, and emissions per driver. To deal with the aforementioned problem and minimise the amount of wasted time, fuel, and emissions during parking search, a promising solution is to design, and implement a smart parking application by utulising IoT data. Such a solution can work with existing infrastructure and provide benefits to a wide range of stakeholders - from drivers to car manufacturers, parking space vendors and government.

However, most of the existing research in this domain take only real-time availability of parking facilities into account during the decision-making procedure. For example, if a driver is planning for a trip to a location that is 30 min away from its current location, and the IoT application suggests parking options based on their current availability, that parking might not be available anymore by the time the smart vehicle reaches the destination. Hence, to maximise the potential of such an application, it is vital to predict the future availability/capacity of parking facilities. Therefore, the smart parking application can navigate drivers to the best available parking bay that will be available when the vehicle reaches its destination.

3 Related Work and State-of-the-Art

3.1 Context- and Situation-Awareness

Context is a key characteristic of modern IoT-enabled systems. According to the widely acknowledged definition given by Dey and Abowd [14], context is "any in-formation that can be used to characterize situation of an entity". In plain words, any piece of information that the system has is a part of the system's context. The aspects of context include, but are not limited to, location, identity, activity, time. The system is context aware "if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task". In simple words, the definition means that the system is context aware if it can use the context information to improve its performance, efficiency, effectiveness and utility. Although recognised as an interdisciplinary area, context-awareness is often associated with pervasive computing, and more recently with the Internet of Things (IoT). Context awareness is a core functionality in IoT, and any pervasive computing system is context aware to some extent.

IoT devices have sensing, actuation, computational and storage capabilities. These devices directly measure the environment characteristics (like temperature, light, humidity). Observation can be also considered as direct user input using keyboards, touchscreens, and voice recognition. Sensor information and user inputs are often processed in a similar manner. After highly heterogeneous input data is delivered, the first processing step is the data fusion and low-level validation of sensor information. Sometimes raw sensor data, collected in a single vector of values, are already viewed as low-level context. The distinction between different levels of context depends on the amount of pre-processing performed upon the collected sensor information. Usually raw or minimally pre-processed sensor data is referred to as low-level context, while the generalized and evaluated information is referred to as high-level context [15].

The situation awareness in pervasive computing and IoT can be viewed as the highest level of context generalisation [16]. Situation awareness aims to formalise and infer real-life situations out of context data. From the perspective of a context aware IoT system, the situation can be identified as "external semantic interpretation of sensor data", where the interpretation means "situation assigns meaning to sensor data" and external means "from the perspective of applications, rather than from sensors" [15]. Therefore, the concept of a situation generalises the context data and elicits the most important information from it. Properly designed situation awareness extracts the most relevant information from the context data and provides it in a clear manner.

3.2 Prediction Techniques

Context prediction aims to predict future context information. It can be done on any level of context processing, starting from low-level context prediction and ending with situation prediction. The existing prediction techniques which can be adapted to context prediction include [17]:

Sequence Prediction Approach. This approach to context prediction is based on the sequence prediction task from theoretical computer science and can be applied if the context can be decomposed into some kind of event flow.

Markov Chains Approach. Context prediction techniques based on Markov chains are quite widespread. Markov chains provide an easily understandable view of the system and can be applied if the context can be decomposed into a finite set of non-overlapping states.

Bayesian Network Approach. This can be viewed as the generalisation of the Markov models. It provides more flexibility but requires more training data in turn.

Neural Networks Approach. Neural networks are biologically inspired formal models that imitate the activity of an interconnected set of neurons. Neural networks are quite popular in machine learning. Context prediction approaches based on neural networks are extensively used and perform well.

Branch Prediction Approach. This approach initially comes from the task of predicting the instruction flow in a microprocessor after the branching command. Some context prediction systems use similar algorithms.

Trajectory Prolongation Approach. Some context prediction approaches treat the vector of context data as a point in multidimensional space. Then the context predictor approximates or interpolates those points with some function, and that function is extrapolated to predict future values.

Expert Systems Approach. Based on expert systems and rule-based engines, the expert systems approach appears in some works on context prediction. The goal of the approach is to construct the rules for prediction. It provides a clear view of the system.

One of the context prediction research challenges is the development of a general approach to context prediction. Many context prediction approaches were designed to fit a particular task and most of them were not designed to be generaliseable (although some of them have generalisation capability). The context prediction process consists of several steps [18]:

Sensor Data Acquisition. This step takes data received from multiple sensors and arranges them into the vector of values. Feature extraction. This step transforms raw sensor data for further usage. From vector of sensor data, vector of features is formed.

Classification. Performs searches for recurring patterns in context feature space. Growing neural gas algorithm was considered to be the best choice. The result of the classification step is a vector of values that represents degrees of membership of a current vector to a certain class.

Labelling. This is the only step that involves direct user interaction. The frequency of involvement depends on a quality of clustering step if classes are often overwritten and replaced that will result in more frequent user involvements.

Prediction. This step takes the history of class vectors and estimates a future expected class membership vector. Context prediction is a relatively new problem for computer science research. The area of context prediction is just being developed and still there are numerous challenges yet to be addressed. Those challenges include [17]:

Lack of General Approaches to the Context Prediction Problem. Most current solutions predict context for particular situations. There have been only a few attempts to define and solve the context prediction task in general.

Lack of Automated Decision-Making Approaches. Most context prediction-related works focused the efforts on prediction itself, but proper acting on prediction results usually was not considered. Most context prediction systems employed an expert system with pre-defined rules to define the actions based on prediction results. With one notable exception of Markov decision processes, almost no systems considered a problem like "learning to act".

Mutual Dependency Between System Actions and Prediction Results Is Not Resolved. This challenge is somewhat related to the previous one. Many context prediction systems considered the tasks of predicting the context and acting on predicted context in sequence: predict and then act on prediction results. That approach can handle only simplified use cases when actions do not affect prediction results. For example, in a smart home the system can employ any policy for switching the light or opening the door in advance, depending on user movement prediction results. But whatever the system does, it will not affect user intentions to go to a particular room. However, in a general case system, actions do affect prediction results. For example, consider a system which is capable of automatic purchases to some degree and which needs to plan the expenses, or in a more serious use case, consider a pervasive system that is capable of calling the ambulance and that needs to decide whether to do it or not depending on observed user conditions. In those and many more use cases, prediction results clearly will depend on what the system does. However, there are almost no work that considered the problem of mutual dependency between system actions and prediction results. So far, the only works that did address that problem were the ones on the Markov decision processes as discussed above. The task of resolving that dependency is actually a special case of a reinforcement learning task. In our opinion, although comparing to most reinforcement learning task pervasive computing systems have their own specifics (e.g., relatively obscure cost and reward functions, high cost of errors and therefore very limited exploration capabilities), recent advancement in the reinforcement learning area can help to over-come that problem.

If all those context prediction challenges are resolved, it will IoT systems handle more sophisticated use cases, enhance the applicability and the effectiveness of context prediction techniques and therefore enhance overall usability of IoT-enabled context-aware systems.

The next section will present and discuss the proposed Context Prediction-as-a-Service (CPaaS) engine.

4 Context Prediction as a Service—Vision and Open Challenges

As a step towards operationalising context-awareness in the realm of IoT, IoT middleware platforms, also known as Context Management Platform (CMP), have become a significant research challenge. CMPs manage interactions with sources of context (context providers (CP)) and offer contextual information to context-aware applications (context consumers (CC)) as a service. In our earlier research, we have developed a novel CMP called Context-as-a-Service (CoaaS) [19]. As it is illustrated in Fig. 2, CoaaS acts as a middleware that facilitates communication between CC and CP. One of the main features of CoaaS that distinguishes it from other existing CMPs is its generic and flexible query language that allows developers of context-aware IoT applications to query and monitor context of the entities of interest in real-time [3]. More importantly, Context Definition and Query Language (CDQL) supports queries about multiple entities and their situations (i.e., high-level, inferred context), which can be defined as part of the query at run time [3]. Another unique distinguishing feature of CoaaS is continuous reasoning and applying AI over IoT data streams and situations monitoring. Our experimental results show that CoaaS platform

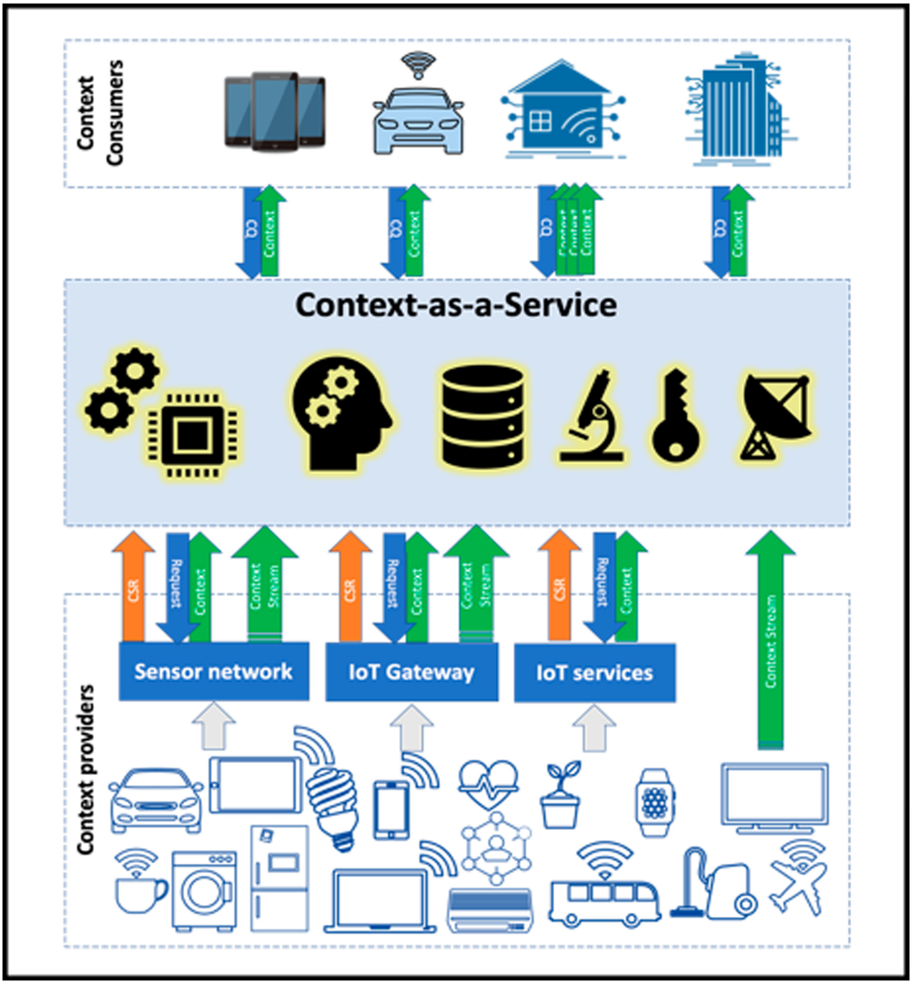

Fig. 2. CoaaS conceptual architecture

has a better overall performance in the execution of complex context queries compared to its main rival Fiware Orion [20] developed as part of EU research projects.

While current version of CoaaS platform can be utilised by context-aware IoT applications to seamlessly acquire and monitor the current context/situation of IoT entities, it does not support predicting future context/situations. IoT applications generally operate in unstable and unpredictable environments, where the context of entities constantly changes. It is insufficient to perform querying and adaptation solely based on the current context. To fully realise the potential benefits of context-awareness, it is essential to provide IoT applications with the capability to predict the future context and situations so they can adapt proactively to future changes, and take preemptive smart actions to mitigate any undesired or negative consequences. Context prediction will also enable applications to make better decisions and manage resources more efficiently.

A CMP that supports context prediction can offer distinct advantages to IoT ecosystems. It will enable a range of intelligent and proactive IoT applications and services in multiple domains. For example, it can enable smart parking with near real-time accurate prediction of parking availability and navigate drivers to available parking spaces hence saving fuel, time, and reducing emissions' impact on the environment.

The existing context prediction research is rather limited in meeting IoT applications' prediction demands. First, they do not support predicting multiple contexts and real-world complex situations that are highly important for operating in dynamic IoT ecosystems. Second, they mostly lack the ability to update their prediction model and suffer from the 'stale' model problem that can degrade accuracy. Third, they generally do not incorporate any actuation mechanism for mitigation purposes. Finally, the major shortcoming in existing context prediction approaches is that they are application-specific, and not applicable and accessible to IoT applications as a general service. The key challenges of context prediction and learning over predicted context in IoT ecosystems are listed below:

- Developing theoretical underpinnings and enhancing the prediction techniques for context/situation prediction, and incorporate them into a library where the best technique can be selected at run time according to the requirements of IoT applications in order to increase the accuracy and efficiency of prediction.
- Developing an advanced learning approach, using the state-of-the-art AI and machine learning techniques, to monitor system evolution and adaptation based on predicted context, and continuously update and extend the existing knowledge and heuristics to increase the prediction accuracy and efficiency.
- Developing a proactive actuation mechanism that can automatically re-evaluate and react to predicted context/situations by recommending actuations and mitigating actions to context consumers (entities and/or applications) that might be affected by the predicted context/situations.

- Developing new query language constructs for context and situation prediction to allow interactions between the CMPs and context consumers in a uniform way.

To address these challenges, in this paper, we propose to extend CoaaS by introducing a new component, called Context Prediction Engine (CPE). CPE will provide a generic mechanism for real-time context prediction, i.e., Context-Prediction-as-a-Service (CPaaS). Accordingly, we will extend the CDQL language with new constructs for context prediction, which can be used to satisfy the needs of context consumers.

5 Context Prediction Engine (CPE)

Significant challenges have to be addressed in researching, advancing, integrating and validating a generic real-time context prediction mechanism to enhance CoaaS and supporting real-time context prediction over IoT data. In [17], we conducted a comprehensive investigation into the context prediction techniques and challenges and identified the essential requirements of such a system while also identifying knowledge gaps in the current literature. Seven prediction approaches have been identified that can be used to predict context of IoT entities. These prediction approaches are Sequence prediction approach, Markov chains approach, Bayesian network approach, Neural networks/deep learning approach, Branch prediction approach, Trajectory Prolongation/Approximation approach, and Expert systems approach. Each of these approaches works best for a certain type of context data and would also depend on other corresponding meta-data, such as what amount of data is available, frequency of observations, and seasonality of the data to name a few. Hence, development of a generic context prediction mechanism, implies a dynamic context prediction mechanism that contains the library of the above-mentioned prediction techniques integrated, and based on the type of context data (e.g., location), the characteristics of data (e.g., observation frequency), and requirements of context provide (e.g., accuracy), applies the matching prediction techniques (e.g., Trajectory prolongation/approximation).

5.1 Problem Definition

To formulate the context prediction algorithm selection problem, consider a library of prediction techniques with n registered prediction algorithms and a given prediction task. The set of prediction techniques is denoted by $P = \{p_1, p_2, \ldots, p_n\}$, and the prediction task is denoted by:

$$pr_i = \langle metaContext_i, metaData_i, requirments_i, contextValues_i \rangle \tag{1}$$

where

$$metaContext_i = \langle entitytype, contextattributetype, fressness, ontology, \ldots \rangle \tag{2}$$

$$requrments = \langle ccuracy, responsetime, predictioninterval, \ldots \rangle$$
$$contextValues = \{c\nu_t, c\nu\nu_{t-1}, c\nu_{t-2}, \ldots\}$$

The goal of a context prediction selection is choosing the most appropriate prediction technique for a given prediction task in a way that the prediction accuracy is maximised. Hence, the problem of context prediction selection can be formulated as an optimization problem:

$$min_j(ca_{t+h}^{entity_i} - p_j(pr_i)), wherep_j \in \{p_1, p_2, \ldots, p_n\} \tag{3}$$

5.2 Context Prediction Engine Framework

Figure 3 presents the architecture of the proposed CPaaS framework, with a specific focus on the context prediction engine (CPE). CPaaS has five main components, namely Communication & Security Manager (CASM), Context Reasoning Engine (CRE), Context Storage Management System (CSMS), Context

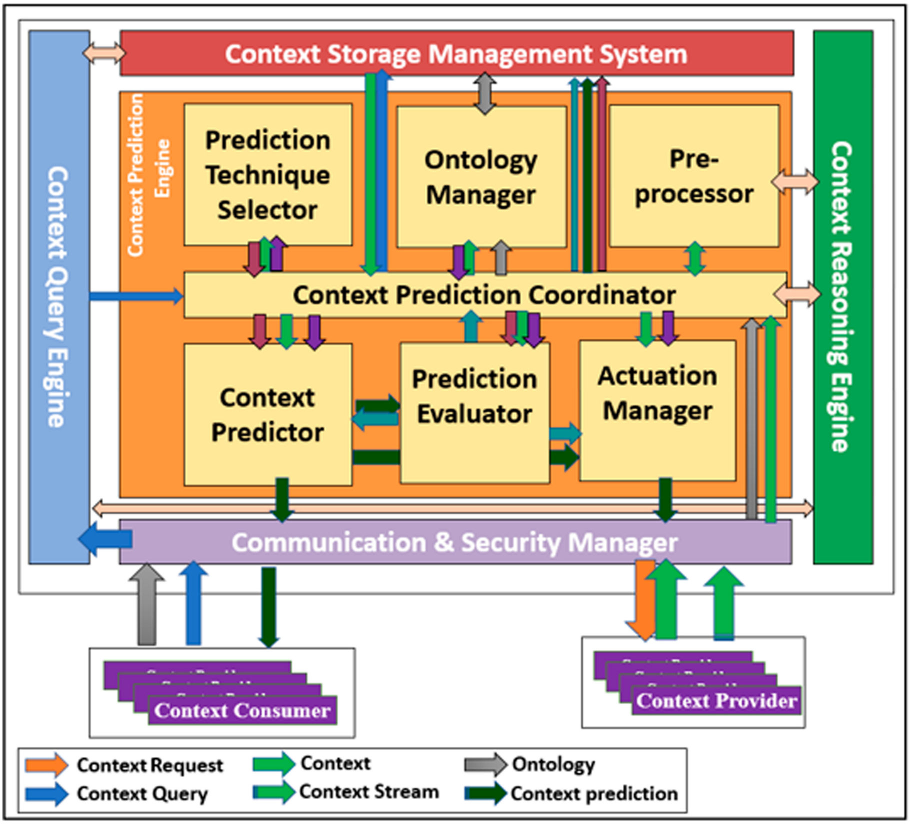

Fig. 3. Context-Prediction-as-a-Service framework

Query Engine (CQE), and Context Prediction Engine (CPE). CASM acts as a proxy and distributes all the incoming messages from CPs and CCs to the corresponding components after performing authentication and authorisation. CQE is responsible for parsing the incom-ing queries, generating and orchestrating the query execution plan, and producing the final query result. CSMS is in charge of storing descriptions of context services and facilitates service discovery, caching contextual information, and storing and analysing the historical context. The main task of CRE is to infer situations from raw sensory data or existing primitive low-level context. The details of these components are available in our earlier publications [21,22].

Lastly, CPE, which is the main focus of this paper is responsible for predicting the future context and complex situations and taking preemptive actions to mitigate any undesired or negative consequences. CPE consists of six main subcomponents, namely, Context Prediction Coordinator (CPrC), Ontology Manager (OM), Prediction Technique Selector (PrTS), Context Predictor (CPr), Prediction Evaluator (PrE), and Actuation Manager (AM).

When a CDQL query with a prediction task is issued to CPaaS, after passing the security checks, it will be sent to the Context Query Engine (CQE) by Communication and Security Manager (CASM). Then, the parsed query plus some additional information, such as meta-data about the context of interest, will be sent to the Context Prediction Coordinator (CPrC).

The CPrC plays an orchestration role in the prediction engine. This module is responsible for managing and monitoring the whole execution procedure of a prediction task. In the next step, prediction tasks will be pushed into OM module. This module is in charge of finding the possible correlated context attributes that can be used to better predict the future context of interest. Moreover, the OM identifies the ontology class of the context entity of interest as well as its context attribute type. CPaaS allows Context Consumers to define and register their own ontology to offer more flexibility to developers of context-aware IoT applications.

After discovering the correlated context attributes, the prediction task is sent to PrTS. PrTS searches the library of available prediction techniques and chooses the most suitable one based on several parameters related to the incoming prediction task, such as type of context, context meta-data, and correlated context-attributes.

The goal of the PrTS is to select the most appropriate prediction technique from a prediction set P that matches a given prediction task pr_i. There are a number of similarity measures such as Euclidean and Manhattan distance measurements, Pearson coefficient measurement, and Cosine similarity measurements that can be used [23]. However, these methods are not able to deal with the impreciseness and vagueness associated with prediction requirements of IoT applications and the uncertainty of context. Fuzzy set theory has been recognized for its strength in modeling imprecise and uncertain information. By using a fuzzy matching method, similarity measurement can take the context uncertainty into consideration, and as a result increase the prediction accuracy. We can represent the fuzzy selection of a prediction algorithm as below:

$$fuzzyselection_m = \{(pr_m, p_m, \mu_{(pr_m,p_m)}), m = 1, \ldots, n\} \quad (4)$$

The membershiop degree of p_m for pr_m is represented by $\mu_{(pr_m,p_m)}$. The fuzzy selection aim is to find a prediction algorithm p_m for the application which has a fuzzy matching with the prediction task pr_m, using a fuzzy rule that implies $pr_m \rightarrow p_m$.

In parallel with the previous step, a request is sent to the Context Storage Management System (CSMS) to fetch the related historical context of the attribute of interest and other correlated attributes. Lastly, all the retrieved information in previous steps is passed to the CPr module.

CPr builds a prediction model based on the provided information and performs the requested context prediction. The outcome of the prediction is sent back to the Context Consumer though the CASM. CPr also caches the prediction result and model for future usage. Moreover, the predicted context/situation is sent to AM.

AM is responsible for identifying the possible actions that need to be taken proactively based on the outcome of the prediction and notifying corresponding context consumers. To achieve this goal, AM uses OM to discover the severity of the predicted situation and possible actions. Then, AM query for all the context consumers that subscribed for the predicted situation and push an actuation signal to them.

After the prediction task is completed, the CPE keeps monitoring the real-time value of the context attribute of interest for a certain amount of time. The motivation behind this procedure is to evaluate the generated prediction model against the new knowledge and update it if needed to improve the prediction accuracy at the next cycle. PrE is in charge of this task. To do so, the real-time value of the context attribute is sent to the CPrC. Then, after performing a pre-processing on the value, it is passed into the PrE. In addition to the cleaned, real-time value of the context-attribute, other relevant information, such as the predicted value(s), generated prediction model, the prediction task, and historical values is also shared with the PrE. Then, PrE measures the accuracy of the prediction engine and tries to enhance the prediction accuracy by applying machine learning techniques, in particular, evolutionary learning.

Here, one of the major drawbacks of discriminative machine learning techniques, such as deep networks, are the lack of explainability and interpretability. More even so since deep networks do not have an inherent representation of causality. Moreover, with the rapid development of autonomous sensing platforms and decision support using deep learning there is an urgent need to add an automated interpretation and identification of the underlying processes and parameter states that govern the predictive behaviour of the network. This is even more important for legal and ethical evaluation and compliance particularly with mission-critical applications. Expressing the knowledge implicit in the network using hierarchical models so as to represent the knowledge in the network in an easily interpretable manner has analogies with mixtures of learners in the literature. This is an important observation since these have an explicit syntactic richness to support the extraction of declarative rules, a property that has been used in syntactical and structural pattern recognition for SVMs and shallow networks. Further, these mixtures of learners have been used in formulations based upon mixture models, Markov Logic Networks, decision trees and Bayesian Networks, all of which provide the ability to extract complex probabilistic relationships and impose constraints on the inference process.

6 Conclusion and Future Work

In this paper a novel Context Prediction Engine was proposed. It supports real-time context prediction and machine learning using deep networks, such as GANNs. Future work is concerned with prototyping, integrating and testing the software components, as well as collaborating with ETSI in further developing and improving standards related to context management platforms.

References

1. van der Meulen, R.: Gartner says 8.4 billion connected 'Things' will be in use in 2017 up 31 percent from 2016. Gartner. Letzte Aktual. **7**, 2017 (2017)
2. ETSI - ETSI ISG CIM group releases first specification for context exchange in smart cities. https://www.etsi.org/newsroom/news/1300-2018-04-news-etsi-isg-cim-group-releases-first-specification-for-context-exchange-in-smart-cities. Accessed 18 Feb 2019
3. Hassani, A., Medvedev, A., Delir Haghighi, P., Ling, S., Zaslavsky, A., Prakash Jayaraman, P.: Context definition and query language: conceptual specification, implementation, and evaluation. Sensors **196** (2019). https://doi.org/10.3390/s19061478
4. Australia bushfires: Which animals typically fare best and worst? - BBC News. https://www.bbc.com/news/world-australia-50511963. Accessed 10 Feb 2020
5. Counting the terrible cost of a state burning. https://www.smh.com.au/national/counting-the-terrible-cost-of-a-state-burning-20090208-811f.html. Accessed 10 Feb 2020
6. Bushfire Emergency. https://www.wwf.org.au/get-involved/bushfire-emergency#gs.wj4zec. Accessed 10 Feb 2020
7. Fox, S., Potts, J.M., Pemberton, D., Crosswell, D.: Roadkill mitigation: trialing virtual fence devices on the west coast of Tasmania. Aust. Mammal. (2019). https://doi.org/10.1071/AM18012
8. Englefield, B., Candy, S.G., Starling, M., McGreevy, P.D.: A trial of a solar-powered, cooperative sensor/actuator, opto-acoustical, virtual road-fence to mitigate roadkill in Tasmania, Australia. Animals **9**(10), 752 (2019)
9. 3218.0 - Regional Population Growth, Australia (2011). https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/3218.02011?OpenDocument. Accessed 10 Feb 2020
10. 3218.0 - Regional Population Growth, Australia, 2017–18. https://www.abs.gov.au/AUSSTATS/abs@.nsf/Lookup/3218.0Main+Features12017-18?OpenDocument. Accessed 10 Feb 2020
11. Australia's capital cities - McCrindle. https://mccrindle.com.au/insights/blog/australias-capital-cities/. Accessed 10 Feb 2020

12. Australian drivers spend over 3,000 hours looking for parking in their lifetime - Parkhound. https://www.parkhound.com.au/blog/australian-drivers-spend-over-3000-hours-looking-for-parking-in-their-lifetime/. Accessed 10 Feb 2020
13. Drivers spend an average of 17 hours a year searching for parking spots. https://www.usatoday.com/story/money/2017/07/12/parking-pain-causes-financial-and-personal-strain/467637001/. Accessed 10 Feb 2020
14. Dey, A.K.: Understanding and using context. Pers. Ubiquitous Comput. **5**(1), 4–7 (2001). https://doi.org/10.1007/s007790170019
15. Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: a review. Pervasive Mob. Comput. **8**(1), 36–66 (2012). https://doi.org/10.1016/j.pmcj.2011.01.004
16. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Hum. Factors **37**(1), 32–64 (1995)
17. Boytsov, A., Zaslavsky, A.: Context prediction in pervasive computing systems: achievements and challenges (2011)
18. Mayrhofer, R.: An architecture for context prediction. Adv. Pervasive Comput. (2004)
19. Moore, P., Xhafa, F., Barolli, L.: Context-as-a-service: a service model for cloud-based systems. In: Proceedings of the 2014 8th International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2014, pp. 379–385 (2014). https://doi.org/10.1109/CISIS.2014.53
20. Fiware-Orion. https://fiware-orion.readthedocs.io/en/master/. Accessed 18 Feb 2019
21. Hassani, A., Medvedev, A., Zaslavsky, A., Haghighi, P. D., Jayaraman, P.P., Ling, S.: Efficient execution of complex context queries to enable near real-time smart IoT applications. Sensors (2019). https://doi.org/10.3390/s19245457
22. Hassani, A., Medvedev, A., Zaslavsy, A., Delir Haghighi, P., Ling, S., Indrawan-Santiago, M.: Context-as-a-service platform: exchange and share context in an IoT ecosystem. In: Percom 2018 (2018)
23. Siddiquee, M.M.R., Haider, N., Rahman, R.M.: A fuzzy based recommendation system with collaborative filtering. In: SKIMA 2014–8th International Conference on Software, Knowledge, Information Management and Applications (2014). https://doi.org/10.1109/SKIMA.2014.7083524