Towards Context Classification and Reasoning in IoT

Abayomi Moradeyo Otebolaku
Dept. of Computer Science
Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool, United Kingdom
a.m.otebolaku@ljmu.ac.uk

Gyu Myoung Lee
Dept. of Computer Science
Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool, United Kingdom
g.m.lee@ljmu.ac.uk

***Abstract*— Internet of Things (IoT) is the future of ubiquitous and personalized intelligent service delivery. It consists of interconnected, addressable and communicating everyday objects. To realize the full potentials of this new generation of ubiquitous systems, IoT's 'smart' objects should be supported with intelligent platforms for data acquisition, pre-processing, classification, modeling, reasoning and inference including distribution. However, some current IoT systems lack these capabilities: they provide mainly the functionality for raw sensor data acquisition. In this paper, we propose a framework towards deriving high-level context information from streams of raw IoT sensor data, using artificial neural network (ANN) as context recognition model. Before building the model, raw sensor data were pre-processed using weighted average low-pass filtering and a sliding window algorithm. From the resulting windows, statistical features were extracted to train ANN models. Analysis and evaluation of the proposed system show that it achieved between 87.3% and 98.1% accuracies.**

***Keywords*—IoT, context awareness, context sensing, context recognition.**

I. INTRODUCTION

The IoT, an emerging global Internet based information architecture, is the future of ubiquitous sensing and personalized service delivery as it promises a new world where all objects around us are connected to the Internet, having the capability to communicate with each other with minimal human interventions [1-2]. IoT allows people and things to connect anytime, and anywhere, with anything and anyone, ideally using any network and any service [3]. It is the new Internet where things and humans become addressable and readable counterparts [11]. Thus, physical and virtual objects can cooperate in social interactions, where each entity can produce or consume intelligent services [13]. With this latest revolutionary development, it is now possible for our everyday objects to understand our needs: what we want or prefer, where and when we need them.

To realize such capability, IoT platforms should incorporate intelligent functionality such that IoT objects can understand one another and the environments in which they are situated to move from the realm of smart objects to more practical realm of smart communities. Equipped with sensing and data processing capabilities, IoT infrastructure will incorporate the necessary functionality to sense, pre-process and extract high-level context knowledge from raw sensor data. This will foster the enrichment as well as understanding of human-to-object or object-to-object interactions, ultimately enhancing the anywhere, anytime delivery of intelligent services, which are tailored to our interests [21].

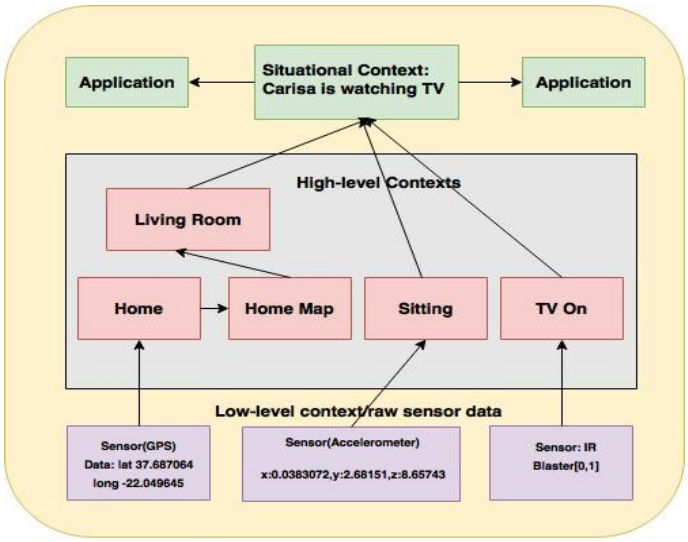

Fig. 1. Situational context from High-level context and low-level context data

So far, data obtained directly from smart objects are unreliable for delivering personalized intelligent services because these data contain errors and noise. Thus, contexts obtained from such objects will be biased, and might lead to misleading conclusion. To address this problem, these raw context data can be processed to obtain meaningful high-level context information. The solution consists equipping IoT objects with the capability to infer objective as well as subjective context information from trustworthy sources to characterize an agent such as a user, a service or an object to provide intelligent and credible personalized services in the IoT environments. The objective contexts, on the one hand, are context data such as location, time, illumination, noise, etc. coming from physical sensors, characterising the situations of agents. On the other hand, the subjective context captures cognitive data such as preferences, trust, etc. of the agents [11].

A contextual situation or situational context characterising any agent is at higher semantic level than high-level context. Contextual situations are products of semantic relationships between high-level contexts or between low-level and high-level contexts. It is a context information that is expressed in a way similar to how humans express "conditions". In [4], Ye et al. defined situational context as follows: " *Situation can be defined by collecting relevant contexts, uncovering meaningful correlations between them, and labelling them with a descriptive name*". IoT objects should be capable of identifying situational contexts. This type of contexts is more meaningful and can be easily interpreted by humans and applications. For example, let us assume that a user, *Carisa,* is located at home, and that she is sitting in the living room. Now, suppose that the TV in the living room is switched on we can conclude that Carisa is watching TV! "*Carisa is watching TV"* is a situational context. Similarly, "*located at home"*, "*sitting in the living room"*, and "*TV is on"* are situational contexts obtained from combinations of primary contexts as illustrated in Fig.1. Thus, "*sitting"," home", "TV", "living room"* (which can be obtained from the map of the "*home*") are high-level contexts, gleaned from raw sensor data.
The figure also illustrates the transition from raw sensor data to high-level context information.

Presently, not all IoT context platforms provide such functionality for obtaining high-level and situational contexts from raw IoT sensor data as illustrated above. Thus, until IoT platforms are able to classify such high-level contextual information from raw data to characterize real or virtual entities in the real world, only then can context information be successfully shared between IoT objects for delivering intelligent services. This kind of new context awareness will extend the traditional notion of context from "*any information that can be used to characterize the situation of an entity, e.g. person, place object, etc. is considered relevant to the interaction between a user and an applications themselves*" [20] to a context model of the real world that can be shared and reused across heterogeneous and interconnected objects. Examples of sensors are accelerometers and gyroscopes, motion sensors used for sensing movements such as acceleration, velocity, etc.. For example, GPS sensor provides raw location data as coordinates such as +37.687064, -22.049645, which do not provide us with any meaningful information about the physical address such coordinates represent.

However, as we know, a lot of information such as locality, city, country, etc., can be derived from those two coordinates. Similarly, take these x, y, z (-0.0383072, 2.68151, 8.65743) of a tri-axial accelerometer, these data provide no clue that can lead us to knowing that the user was driving or walking. Essentially, raw sensor data are aggregated, and extracted as useful data features, which are then fed into classification models to derive meaningful high-level contextual information

This paper presents the design of a framework for context awareness in the IoT, based on ANN model as an important component of our existing IoT platform to extract high-level contexts from objects' raw context data. Using this ANN model, we evaluate the sensitivity of the proposed system to classification of high-level contexts from raw sensor data.

The rest of the paper is organized as follows. Section II presents background and related work. In section III, we present details of the proposed context classification framework. In section IV, we analyze and present ANN as model for classifying context raw sensor data. Section V presents our experiments and evaluation results. In section VI, we draw conclusion and outline our future work.

II. RELATED WORK

IoT possesses the potentials to take context-awareness to the next level considering that massive data coming from diverse sensors can be explored to build intelligent applications [11]. Context awareness has been extensively explored as a fundamental and key feature of mobile computing systems for adaptive decision making [22]. Therefore, the outcomes of reasoning about information derived from context data should be explored to enhance user's perceived quality of personalized services in ubiquitous environments [21]. However, majority of current IoT systems focus only on raw context data. For example, [16] presents an architecture that enhances the context-aware capability of IoT middleware solutions, enabling to build a sensing-as-a-service platform.

Similarly, Hussein et al. [11] proposed a system for semantic context awareness in IoT with novel concepts of objective and subjective context information in social IoT environments. Nevertheless, this solution only assumes the availability of classification of high-level contextual information from raw sensor data. Mingozzi et al. [17] also present a semantic-based context model for quality of service support in IoT, where a semantic ontology model has been developed for context reasoning to infer high-level context information for allocating services to applications while meeting QoS requirements.

Perera et al. [2] present a more comprehensive survey of context awareness in IoT, with detailed analysis and

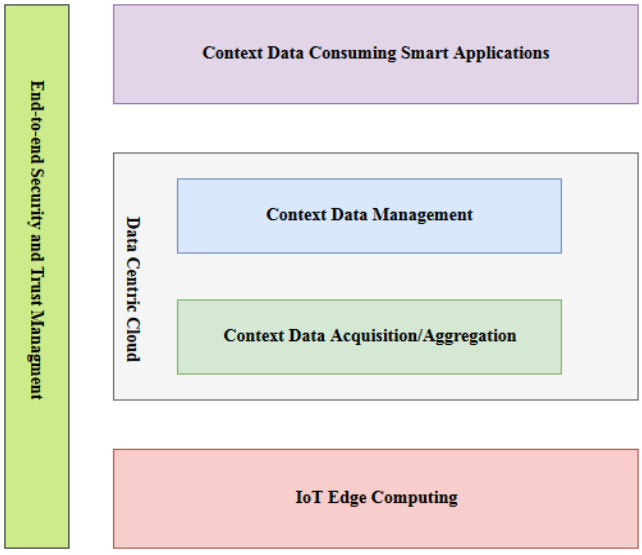

Fig. 2 . IoT Context Abstraction Layers

information. One of the key requirements that need more attention is the capability of these systems to identify and classify high-level context information from raw context data. Some other platforms, such as FIWARE with its context broker, provide the capability to obtain low-level context data from IoT sensors as well as providing interfaces for applications to access these data. However, how to classify high-level contexts from these data remains an open issue [15].

These proposals are just few of numerous ongoing work addressing context recognition and reasoning in IoT environments. Nevertheless, unlike these existing proposals, we propose an IoT based context recognition architecture that can be incorporated into our trust evaluation management (TME) system as a key functionality [19]. Our goal is to develop and integrate IoT context recognition framework, which incorporates trustworthiness as depicted in Fig. 2 and Fig. 3. In the current article, however, we focus on the high-level architecture of our platform. As an important component of our trust-based IoT platform [20], we also present results of the evaluations of context recognition processes developed.

III. Proposed System Framework

A. Overview of the proposed System Architecture

The predicted deployment of billions of devices and their global access require layered management architecture from the physical devices (edge layer) to the cloud computing [18]. In this section, we introduce a high-level overview a typical IoT platform, such as the one being developed by the Wise-IoT project consortium [18]. Wise-IoT project is an H2020 funded project, which is being executed by leading European research and academic institutes and their counterparts in South Korea. Wise-IoT aims to provide a worldwide interoperable Internet-of-Things that utilizes a large variety of different IoT systems and combine them with contextualized information from various data sources.

The IoT platform is broadly a three-layer architecture as shown in Fig. 2. Thus, the first layer of the architecture is the sensor or physical layer where many heterogeneous IoT devices are deployed. These devices can communicate with each other, using standardized connectivity such as Wi-Fi, 3GPP, 3G/4G, Bluetooth, ZigBee and LoRaWAN (supporting low power connectivity) based on such standards as OneM2M. The OneM2M standard is a global IoT standard that addresses communication issues for a common IoT service layer and interoperability on the IoT connectivity layer [15] [18]. This sensor layer is also responsible for the IoT management.

The second layer is the contextual information management layer and can be considered as the Cloud layer responsible for the acquisition, aggregation and processing of context data. Additionally, low-level context data from the physical layer is preprocessed for inference purposes using various techniques such as data mining or machine-learning algorithms as well as Semantic Web models for context reasoning, consisting of a context knowledgebase, query and inference engines. One of the current and popular implementations of this layer is FIWARE context broker, which is based on NGSI standard [15]. Nevertheless, presently these context brokers, such as

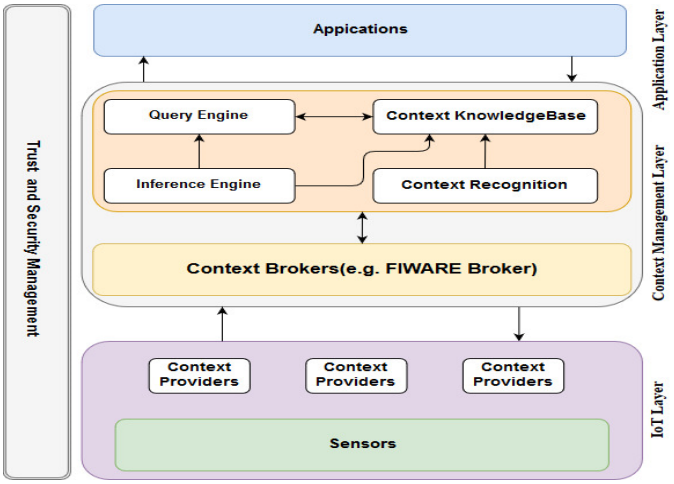

Fig. 3. High-level Architecture of the Proposed System

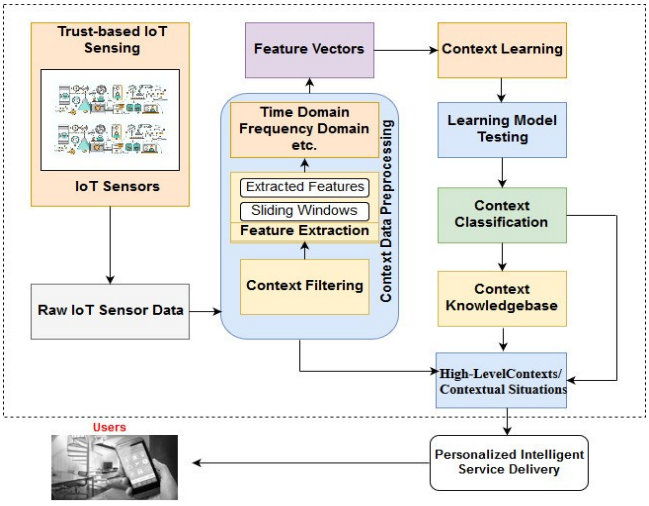

Fig. 4. Context Recognition Processes

FIWARE Context broker, do not provide the capabilities for context classification and knowledge processing and logical inference [15].

The last layer is the application layer where context information is explored to deliver intelligent services. Different kinds of applications can be deployed at this layer, for example, car parking or route recommendation applications.

In the proposed platform, to address issues related to security and trustworthiness of interactions and data exchange between service providers and consumers, an end-to-end security and trust evaluation component is being developed as a cross layer service.

B. Context Recognition Framework

This section introduces the context awareness component as illustrated in Fig. 4 that we are developing, for deployment on IoT platform to process context data obtained from IoT context brokers and other IoT platforms such as FIWARE broker. Each of these components and the algorithms they implement will be discussed next. The framework consists of various processing modules to be incorporated into our trust-based platform for

intelligent service delivery in IoT environment. The context framework has been designed for providing the capabilities for high-level context classification, using machine learning and ontologies for context classification, semantic processing and reasoning.

(i) Context sensing

In the proposed IoT system, context sensing is defined as the acquisition of trusted raw context data in the IoT environment from smart objects, via context broker to characterize the real-world situations. A trusted sensor provides trusted context data. For example, in a room with 5 thermometers, if 4 sensors provide 25 ºC as the temperature readings and one of them provides 50 ºC, then it can be concluded that this sensor cannot be trusted. Additionally, data from IoT objects are error prone. In their raw forms, they are not suitable for building real-world applications. Noise, drifts, and delays are some of the common sources of sensor data errors [12] [14]. In order to mitigate the influence of these errors and noise in the raw data, since they can corrupt the captured context information and consequently the inferred contextual situation, raw data filtering must be performed. This is necessary because applications utilizing context information obtained from these sensors have no control over the outputs of the sensors. However, the filtering process can be used to eliminate or minimize these errors before they are used to infer high-level context information. Also, note that in the architecture, the sensor data should be obtained from trusted IoT sources. The evaluation of such system using trust indicators such as experience, reputation, knowledge, etc. has been reported in [19], and would not be discussed further in this paper.

(ii) Filtering

In our framework, we assume the availability of various kinds of sensors (e.g. accelerometer, thermometer, light, noise, gyroscope, etc.) to design an efficient approach to recognize contexts in an IoT environment. Since raw data from these IoT sensors are prone to high frequency noise and errors, it is important to mitigate the influence of these factors on the collected before using them to classify context information. The signals from these sensors are streamed through a low pass filter to eliminate the high frequency related noise to smoothen the signals. The fundamental concept of a typical low-pass filter is to simply replace the values of a sample by weighted moving average calculated such that $x_l = x_{i-1} \times \alpha + (1-\alpha) \times x_i$ where x_l is the filtered value, x_{i-1} is the previous value and x_i is the current value. This type of filtering is called a weighted moving average because it smoothens the sensor data by replacing each data point with an average of neighboring data points within a given order of the filter [9]. This low-pass filter process produces a response given as the difference equations (1) for a tri-axial sensor, such as accelerometer or gyroscope. For a one- axis sensor such as a light sensor, the filtering is performed on the streams of data it generates.

$$x_{s(i)} = \frac{1}{2N+1}(x(i+N) + x(i+N-1)+, \dots, x(i-N))$$

$$y_{s(i)} = \frac{1}{2N+1}(y(i+N) + z(i+N-1)+, \dots, y(i-N)) \quad (1)$$

$$z_{s(i)} = \frac{1}{2N+1}(z(i+N) + z(i+N-1)+, \dots, z(i-N))$$

$X_s\ (i)$ is the filtered signal value for the i^{th} data point; whereas N is the number of data points on both sides of X_s (i) and 2N +1 is the order of the filter. There are two important advantages for using this method to filter out noise from motion sensors according to [9]. First, while retaining low frequencies, it minimizes random high frequencies in the sensor data. Second, it helps to reduce errors that might have been introduced during context data acquisition.

(iii) Feature extraction

Feature extraction is used as one of the preprocessing techniques to obtain useful hidden information from raw sensor data in order to transform the entire raw data into a useful and reduced representational set of features [5]. Although the raw data from sensors contain lots of hidden information and noise [6], the feature extraction process, if carefully selected, can help to isolate useful features from unwanted ones. Additionally, having redundant features in a large set of data results in high dimensional dataset, which could increase computational requirements of the classification algorithms as well as jeopardizing their recognition accuracies. Therefore, extracting suitable features from the IoT sensor data is a very crucial process to improve the whole context classification efficiency. The feature extraction process as used in this paper is executed in two phases. The first phase corresponds to the process of splitting the sensor data into fixed length of segments or windows. The second phase is the actual process of extracting relevant statistical features from each defined sensor data window.

(a) Sliding window phase

In this first phase of the feature extraction process, an algorithm known as fixed length temporal sliding window or segmentation is used [14]. This algorithm splits the sensor data into data segments of fixed intervals of samples called "windows". A window contains a small part of the sensor signal [5], [14]. Each window is "*overlapped*" to form the next window, preserving a proportion of the previously data to be used as the beginning of the next sample [7]-[8]. The formal definition of this process as used in our proposed framework is provided below. Let *f* represent the function that determines the patterns in the event sequences of N IoT sensors with L axes (e.g. x, y, z), matching the user contexts. *f* is defined as follows: $f : S^l \rightarrow A$ where l is the window length of each sensor's axis and A is a set of contexts to be recognized.

Let $s \in S = \{ s_1, s_2, \dots, s_n\}$, and let $a \in A = \{a_1, a_2, \dots, a_n\}$ be defined as the representations of the devices' built-in sensors and contexts to recognize respectively.

Each sequence of events of a sensor s is represented as vector X^s, Y^s and Z^s defined as:

TABLE 1

FEATURE DESCRIPTIONS AND DEFINITIONS

Feature	Feature Description	Formula
Variance (Var)	Defines the average squared difference from the mean of the sensor data over a sliding window.	$var = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$
Median (Med)	Median is the value that separates the higher half of a window the sensor data from the lower half.	$median = l + \left[\frac{\frac{N}{2} - \sum f_0}{f_w}\right] i$
Standard Deviation (STD)	Measures the distribution of the sensor data over a sliding window.	$STD(\delta) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$
Root Mean Square (RMS)	It represents *n* discreet acceleration of the sensor over a sliding window.	$RMS = \sqrt{\frac{x_1^2 + x_2^2, \ldots, x_n^2}{n}}$
Range	This is the difference between the largest and the smallest values of sensor data over a sliding window. It provides the statistical dispersions of the acquired sensor data.	Range = max-min: Where Max = max({f(x_1),f(x_2),…,f(c_n)}) And min = min({f(x_1),f(x_2),…,f(c_n)})

$$\begin{cases} X^s =< x_0^s, \ldots, x_i^s, \ldots > \\ Y^s =< y_0^s, \ldots, y_i^s, \ldots > \\ Z^s =< z_0^s, \ldots, z_i^s, \ldots > \end{cases} \quad (2)$$

These vectors represent the readings from x, y and z axes' events respectively at time *i*.

Let function f_1 take as input $N \times L$ sequence of sensor data to produce as output K vectors of features Fi. Each vector is labeled with activity a ∈ A = $\{a_1, a_2, \ldots, a_n\}$. Let *l and i* denote windows parameters, representing the window's length and the timestamp when the first window begins respectively. $i+l$ is the total time for the first window and it marks the time when the next window begins. Let *r* be the length of the windows slide. For a 50% windows slide, $r = 0.5\ l$.

Using these definitions, function f_1 is defined as follows in equation (3):

$$W_i : X^s_i \rightarrow M_i, \quad f_1 : M_i \rightarrow F_i \quad (3)$$

Where $M_i = <M_{xi}, M_{yi}, M_{zi}>$ are matrices of temporal groups of sequences of events for each sensor in x, y, and z axes. And W_i is the sequence of events segmented into *d* samples of temporal domain windows or time slices of *l* seconds in length for contiguous readings of the sensor's x, y, z axes respectively, starting at the time *i* as follows:

$$\begin{cases} w_{xi}^s =< x_i^s, \ldots, x_{i+l-1}^s, \ldots > \\ w_{yi}^s =< y_i^s, \ldots, y_{i+l-1}^s, \ldots > \\ w_{zi}^s =< z_i^s, \ldots, z_{i+l-1}^s, \ldots > \end{cases} \quad (4)$$

The window slide *r* defines the next temporal windows as follows:

$$\begin{cases} w_{xi+r}^s =< x_{i+r}^s, \ldots, x_{i+rl-1}^s, \ldots > \\ w_{yi+r}^s =< y_{i+r}^s, \ldots, y_{i+rl-1}^s, \ldots > \\ w_{zi+r}^s =< z_{i+r}^s, \ldots, z_{i+rl-1}^s, \ldots > \end{cases} \quad (5)$$

For each window, the segments that start at time *i* are grouped into the matrix:

$$\begin{cases} M_{xi} =< w_{xl}, \ldots, w_{xl}^N, \ldots > \\ M_{yi} =< w_{yl}, \ldots, w_{yl}^N, \ldots > \\ M_{zi} =< w_{zl}, \ldots, w_{zl}^N, \ldots > \end{cases} \quad (6)$$

(b) Statistical Feature extraction phase

The second phase of the feature extraction process involves generating a set of statistical features from each window known as feature extraction [5] [12]. Following the sliding windowing process explained in the previous section, statistical features are generated from each of the matrices M_{xi}, M_{yi}, M_{zi} to build vector V_i with labeled contexts $a \in A = \{a_1, a_2, \ldots, a_n\}$, of the agent. Therefore, given a set of *n* contexts $A = \{a_1, a_2, \ldots, a_n\}$, every temporal window w_i will produce a vector V_i that is labeled with activity $a \in A$. A function f_2 builds the classifier that learns and finds the mapping between V_i and the context to be identified. In the present work, five statistical features have been defined as shown in Table 1.

IV. CLASSIFICATION USING NEURAL NETWORK ALGORITHM

The evaluation of the proposed system requires the implementation of a predictive model for classifying contextual information in an IoT environment. Thus, the interpretation of sensor's raw data to realize the move to high-level context information is executed in steps as described in section III. The statistical feature vectors generated in the last process are fed into a classifier to produce the predictive model for context classification. This model can be ported onto the IoT platform to infer or extract meaningful contextual information from sensor data. The first part of our design and development of classification model for high-level contexts in IoT is therefore to build a machine-learning model. The second part is to evaluate the developed model. In the current work, we have chosen ANN to build our predictive model. ANN is a flexible algorithm capable of learning non-linear data. It also has the capability to handle reconfiguration, generalized learning and can adapt to errors [6]. Multilayer perceptron algorithm is an ANN that learns a non-linear function $y = f(\sum_i w_i x_i) = f(w^T x)$. Where x and w represent the input vectors and the weight vectors respectively. f is the activation function as shown in Fig. 5. Learning is done by training a dataset, in our case, represented vectors of features extracted from the collected sensor data as a given set of input vectors [X = x_1, x_2, x_3,…, x_m] with a target y_i. Where X represents the feature vector obtained from the processes in section III and $y_1 \ldots y_n$ represent the classes or outputs. The generic feedforward neural network

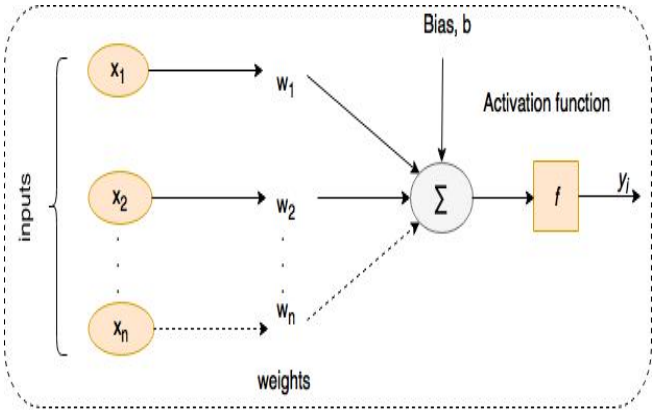

Fig. 5. ANN Neuron

architecture (topology) consists of layers as shown in Fig. 6. illustrates a generic multilayer perception with input, hidden and output layers. The left most block of Fig. 6. represents the input layer consisting of neurons$\{x_i|x_1, x_2, x_3, \ldots, x_m\}$, which represent the inputs to the classifier.

The next block is the hidden layer, which transforms the input values from the previous layer using a weighted summation $\{w_1x_1, w_2x_2, w_3x_3, \ldots, w_mx_m\}$ and the last is the output layer, which transforms the output from the last hidden layer into a final output, which determines from the input the high level contexts represented by such raw data. However, in practice, finding the topology of ANN remains a challenge.

Nevertheless, with series of experimentation, an optimum number of feed-forward network nodes with a non-linear sigmoid activation function $f = \frac{i}{1+e^{-x}}$ has been used in the hidden layer. Note that the output layer contains the number of nodes equivalent to the number of classes (contexts) to be classified. Thus, the two ANN models consists of 7 and 10 output layer nodes respectively. Similarly, the number of input nodes depends on the dimensions of the vector space. Therefore, in the model, there are 30 input nodes for the 2 selected sensors each with three axes. To evaluate the sensitivity of the developed ANN based context recognition model, we used confusion matrix from which we calculate the recall (sensitivity) of the model. We also want to understand, from the confusion matrix, which contexts can be correctly or wrongly classified.

V. EVALUATION OF CONTEXT CLASSIFICATION USING ANN

A. Experimental setup and Raw Data

In order to evaluate the context recognition capability of the proposed framework, we used our existing data [12]. Raw sensor data, which consist of more than 250 thousand records, were collected, labelling various contexts at different locations such as home, office, in bus, train, etc. On the one hand, some contexts were labelled without any indication of the locations (GPS coordinates) where they have been collected. This set of data we call simple contexts. On the other hand, another set of data with labelled with GPS coordinates. This set of data we refer to as complex data. These data were collected from two traditional 3D sensors namely accelerometers and gyroscope. We preprocessed the data using the filtering process as described in section III (B) to mitigate noise and errors. Based on the temporal sliding window algorithm, we extracted five important statistical features as illustrated in Table 1. Thus, with 5 features and 3 dimensions for each of the two sensors, the models, which were implemented using python machine-learning toolkit, scikit-learn [16], have 30 input neurons.

For the output layer, the number of contexts (classes) we wanted to identify determined the number of neurons (7 and 10 output nodes, for each type of contexts: simple and complex). For the hidden layer, after series of experiments, we chose 10 as the optimal number of neurons.

Designing the model this way, the aim was to understand various consumption preferences of users when in different locations and performing various activities so that contextual situations of the agents can be inferred and used by applications to deliver intelligent services tailored to their tastes and preferences. Our target is to develop an IoT platform providing intelligent support to deliver personalized services. However, in this paper, as earlier stated, our goal is to identify such contexts using IoT devices.

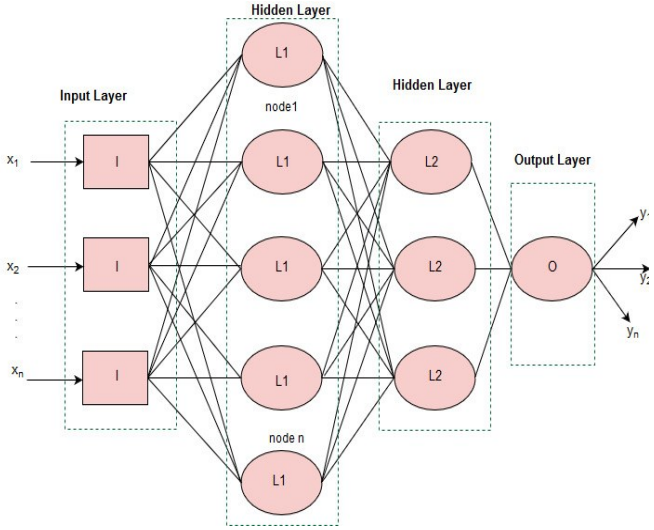

Fig. 6. Our Neural Network Architecture for Context Classification

B. Evaluation Model's Sensitivity

To evaluate the performance of the proposed system, its recognition accuracy and the sensitivity (recall) were computed using confusion matrices of simple and complex context information. Simple contexts are those contexts characterizing the agents (e.g. users) in ubiquitous environment without considering the labelled GPS coordinates of the location of the agent. On the other hand, the complex contexts include labelled GPS in the classification process. Thus, in the first set of experiments, we evaluated the sensitivities of the models to 10 complex contexts and 7 simple contexts. We therefore trained two different models with 10 and 7 nodes representing the context classes respectively. We used two separate datasets and

TABLE 2

CONFUSION MATRIX for SIMPLE CONTEXT EVALUATION

Predicted Class								Actual Class
A	**B**	**C**	**D**	**E**	**F**	**G**		
1673	1	1	1	1	20	1	A	
3	**310**	0	7	2	1	0	B	
0	2	**566**	7	4	0	1	C	
3	0	2	**456**	4	0	1	D	
6	5	10	4	**2177**	11	0	E	
25	6	0	1	16	**2319**	3	F	
2	0	0	0	2	2	**345**	G	

Legend: A = Lying;B = Driving;C = Running; D = Jogging;E = Walking; F = Sitting; G = Standing

adopted the leave-one-subject out testing approach. This means using data from different subjects to train the model and dataset from completely different subjects as testing data. One of the datasets is based on the complex contexts and the other is based on the simple contexts. The importance of this approach is to avoid overfitting. It also helps to understand how the models would behave in real life in terms of generalization. We computed the confusion matrices for both ANN models. The results for simple and complex contexts are shown in Tables 2&3, where the diagonals of the matrices represent the actual number of context correctly predicted by the models. We use (7) and (8) to compute recall and accuracy of the models

$$Sensitivity = \frac{TP}{TP+FP} \quad (7)$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (8)$$

Where TP, TN, FN and FP are true positive, true negative, false negative, and false positive respectively. Using (7), the recall of the model was computed from the confusion matrices in Tables 2 and 3.

Where TP is the number of correctly classified classes and FP is the number of classes incorrectly predicted.

For example, in Table 2, the sensitivity of the model to *lying* context is 98% with the total number of lying context recognized was 1673 but classifying lying 25 times as sitting. The highest classification errors occurred between *lying* and *sitting*. On the other hand, the evaluation of the complex contexts shows a similar trend in the number of misclassifications as observed in the simple context evaluation. However, the number of misclassifications increased as shown in Table 3. For example, looking at the same *lying/Home* and *standing/Train* contexts with location (in Table 3), they have been misclassified by the model 47 times compared to 2 misclassifications as shown in Table 2. The sensitivity of the same lying context with location label dropped to 97%. This result suggests that including location label confuses the models more than when location was not included.

Using (8), the overall performance of the model, in terms of effectiveness, was computed as accuracy from the confusion matrices in Tables 2 &3 for both simple and complex contexts.

TABLE 3

CONFUSION MATRIX FOR COMPLEX CONTEXT EVALUATION

Predicted Class											Actual Class
A	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***		
2129	5	3	8	3	4	5	25	6	4	*A*	
4	**321**	13	4	2	2	9	26	4	8	*B*	
8	8	**341**	18	0	4	6	47	2	18	*C*	
1	2	32	**167**	8	2	1	20	2	2	*D*	
2	3	3	16	**60**	0	0	12	0	5	*E*	
3	1	6	0	0	**190**	0	10	9	0	*F*	
7	13	4	3	1	1	**240**	16	3	3	*G*	
23	19	35	11	7	5	9	**980**	10	9	*H*	
5	6	0	1	0	12	8	9	**257**	4	*I*	
7	12	33	8	4	2	1	30	9	**234**	*J*	

Legend: A=lying/Home
B=Standing/Bus;C=Standing/Train;D=Ascending/Elevator
E=Descending/Elevator; F=DescendingStairs/Office;G=Sitting/Bus
H=Sitting/Home; I=AscendingStairs/Office;J=Sitting/Train

In the case of the latter, the accuracy was 87.3%. Whereas in the former, the accuracy was 98.1%.

C. Impact of Features on the ANN Context Classification

In the last experiments, we evaluated the sensitivity of the MLP classification model. One of the most important factors for determining the sensitivity and accuracy of classification models is the kind of features fed as inputs to train the model. In some cases, there is no need for using many feature vectors in the classification processes. In fact, this can also lead to curse of dimensionality. In this experiment, we decided to evaluate the accuracy of the MLP model for each feature vector. As presented in section 3, five temporal statistical features were extracted from the sensor data. Thus, we evaluated the performance of each feature with respect to the 7 simple contexts. The goal of this experiment is to determine if some of the selected features individually are sufficient to classify these contexts. This knowledge is crucial; it would help to avoid the use of redundant features. Table 4 shows the capability of each of the five temporal features to serve as inputs to the classification model to classify seven contexts (*A: Lying, B: Driving, C: Running, D: Jogging, E: Walking, F: Sitting, G: Standing*).

TABLE 4

ANALYSIS OF STATISTICAL FEATURES

Feature / Context	Var	Med	STD	RMS	Range
A	√	×	×	√	√
B	×	×	√	×	×
C	√	×	√	√	√
D	√	×	√	√	√
E	√	√	√	√	√
F	√	√	√	√	√
G	×	×	×	√	√

The result shows that we do not really need all the five features as inputs to the model for context classification. In fact, we only needed a combination of two of the features to classify all the contexts correctly. Examples of such features are RMS and STD, which when combined can effectively classify the contexts. However, variance (Var) and median (Med) could not identify the entire classes of contexts.

VI. CONCLUSION

In this paper, as part of an ongoing trustworthy IoT based context awareness platform project, we have proposed to incorporate a context recognition process. The work introduced in this paper represents one of the building blocks of the project aiming to develop a trust-based context awareness framework to support intelligent service delivery in an IoT platform. The framework is being developed for integration with our existing trust evaluation platforms [19].

The proposed system provides key functionality for context sensing, preprocessing, filtering, and classification. We have provided an evaluation of the classification functionality of the platform with accuracies between 87.3% and 98.1%. The results also show that rather than using all the feature vectors to train the predictive model, two statistical feature vectors can be used effectively to classify new contexts.

Some potential practical applications among others of the proposed framework are route recommendation systems, car parking, and mobile learning. In mobile learning for example, contexts and activities of learners can be determined using IoT sensors. This kind of information can then be used to learn the preferences of the learners in order to provide relevant learning content according to their contexts and preferences.

In the future, we would evaluate the computational requirements of the context classification model considering the resource requirements such as execution time. We would also like to provide the integration of the context classification platform into our trust evaluation and analysis platform for deciding what data sources should be considered trustworthy when collecting the sensor data.

ACKNOWLEDGMENTS

The work reported in this paper is supported by the European Union's H2020 Programme for research, technological development and demonstration under grant agreement No 723156, the Swiss State Secretariat supported this research for Education, Research and Innovation (SERI) and the South-Korean Institute for Information & Communications Technology Promotion (IITP).

REFERENCES

[1] B. Guo, "From the Internet of Things to Embedded Intelligence", Proc. World Wide Web, vol. 16, pp. 399-420, 2013.

[2] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos "Context aware computing for the internet of things: a survey". IEEE Communications Surveys Tutorials 2013; PP (99):1–41.

[3] P. Guillemin and P. Friess, "Internet of things strategic research roadmap," The Cluster of European Research Projects, Tech. Rep., September 2009, http://www.internet-of-things-research.eu/pdf/IoTCluster Strategic Research Agenda 2009.pdf [Accessed on: 2017-01-05].

[4] J. Ye, S. Dobson and S. McKeever. "Situation identification techniques in pervasive computing: A review. Pervasive and Mobile Computing", 8 (2), pp. 36–66, 2012.

[5] F. Davide, C. Pedro, R.F. Diogo, & J. M. Cardoso," Preprocessing Techniques for Context Recognition from Accelerometer Data. Personal and Ubiquitous Computing", 14(7), 645-662, 2010.

[6] M. Abdullah, A. Negara, S. Sayeed and K. Muthu," Classification Algorithms in Human Activity Recognition Using Smartphones", International journal of Computer and Information Engineering, Issue 6, pp. 422-429, 2012.

[7] O.D.Lara, M.A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors. Communications Surveys & Tutorials", IEEE, vol.15, no.3, pp.1192, 1209, Third Quarter 2013,doi: 10.1109/SURV.2012.110112.00192, 2013.

[8] P. Siirtola and J. RöninG. "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data". International Journal of Interactive Multimedia and Artificial Intelligence, 1 (5): 38-45, 2012.

[9] A.M. Khan, Y-K. Lee, S.Y.Lee, and T-S. Kim, "A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer", Information Technology in Biomedicine, IEEE Transactions on, vol. 14, no. 5, pp. 1166-1172, Sept, 2010.

[10] A. Maarala; X. Su; J. Riekki, "Semantic Reasoning for Context-aware Internet of Things Applications," in *IEEE Internet of Things Journal* , vol.PP, no.99, pp.1-1

[11] D. Hussein, et al. "Dynamic Social Structure of Things: A Contextual Approach in CPSS." IEEE Internet Computing Magazine, vol. 19, no. 3, pp. 12-20, May. 2015.

[12] A.M. Otebolaku, & M.T. Andrade, "User context recognition using smartphone sensors and classification models". Journal of Network and Computer Applications, 66(B), pp.33-51, May 2016.

[13] T.W. Um, G.M. Lee, J.K. Choi, "Strengthening trust in the future Social-Cyber-Physical infrastructure: an ITU-T perspective," IEEE Communications Magazine, ISSN 0163-6804, vol. 54, no.9, pp.36-42, September 2016.

[14] L. Bao and S.S. Intille. Activity Recognition from User Annotated Accelerometer Data. Pervasive Computing, LNCS Vol. 3001, 1-17, 2004.

[15] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall and M. Zhao, "Standards-Based Worldwide Semantic Interoperability for IoT," in IEEE Communications Magazine, vol. 54, no. 12, pp. 40-46, December 2016. doi: 10.1109/MCOM.2016.1600460CM.

[16] Scikitlearn: [last accessed: 14th Jan, 2017] http://scikit-learn.org/stable/modules/neural_networks_supervised.html.

[17] E. Mingozzi, G. Tanganelli, C. Vallati, B. Martnez, I. Mendia, M. Gonzlez-Rodrguez, "Semantic-based context modeling for quality of service support in iot platforms", in: 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM, 2016, pp. 1–6.

[18] Wise-IoT [Accessed on the 23rd, Jan. 2017]:http://wise-iot.eu/en/home/

[19] U. Jayasinghe, N. B. Truong, G. M. Lee and T. W. Um, "RpR: A Trust Computation Model for Social Internet of Things," 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 2016, pp. 930-937.

[20] A. K. Dey, "Understanding and Using Context", *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001.

[21] A.M. Otebolaku and M.T. Andrade "Context-Aware Personalization Using Neighborhood based Context Similarity". Journal of Wireless Pers Commun. Vol. 94 no.3, pp. 1595-1618, June 2017. DOI: 10.1007/s11277-016-3701-2.

[22] A. M. Otebolaku, M. O. Adigun, J. S. Iyilade and O. O. Ekabua, "On Modeling Adaptation in Context-Aware Mobile Grid Systems," *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*, Athens, 2007, pp. 52-52. DOI: 10.1109/CONIELECOMP.2007.90