LEBANESE AMERICAN UNIVERSITY

LP-SBA-XACML: Lightweight Semantics based Scheme embedded with Intelligent Behavior-aware Privacy Preserving Model

By

Mohamad A. Chehab

A thesis
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

School of Arts and Sciences

May 2019

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS APPROVAL FORM

Student Name: Mohamad Chehab I.D. #: 201205174

Thesis Title : LB-SBA-XACML: Lightweight Semantics Based Scheme Embedded with Intelligent Behavior-Aware Privacy Preserving Model

Program: Masters in Computer Science

Department: Computer Science and Mathematics

School: Arts and Sciences

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Masters of Science in the major of Computer Science

Thesis Advisor's Name Dr. Azzam Mourad	Signatu [redacted]	DATE: 3 / 5 / 2019 Day Month Year
Committee Member's Name Dr. Haidar Harmanani	Signatu [redacted]	DATE: 3 / 5 / 2019 Day Month Year
Committee Member's Name Dr. May Hamdan	Signatu [redacted]	DATE: 3 / 5 / 2019 Day Month Year

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

Name: L'emir Mohamad Chehab

Signatur [redacted]

Date: 3 / 5 / 2019
Day Month Year

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU's Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: L'emir Mohamad Chehab

Signature [redacted]

Date: 3 / 5 / 2019
Day Month Year

ACKNOWLEDGMENT

This work would not have been possible without the support of my family and my advisor Dr. Azzam Mourad to ensure producing adequate and high quality work. Moreover, his nonstop guidance and his patience for enduring me paved the way for successfully completing this work.

I am grateful for this opportunity for I got the chance to work with several new technologies and experiment with new ideas and concepts. Such work would have not been possible elsewhere.

I would also like to thank my committee members Dr. Haidar Harmanani and Dr. May Hamdan for their time in revising my work and providing their valuable feedback.

This work is accomplished thanks to everyones help and support.

LP-SBA-XACML: Lightweight Semantics based Scheme embedded with Intelligent Behavior-aware Privacy Preserving Model

Mohamad A Chehab

ABSTRACT

The wide applicability of Internet of Things (IoT) would truly enable the pervasiveness of smart devices for sensing data. IoT coupled with machine learning would enter us in an era of smart and personalized, services. In order to achieve service personalization, there is a need to collect sensitive data about the users. That yields to privacy concerns due to the possibility of abusing the data or having attackers to gain unauthorized access. Moreover, the nature of IoT devices, being resource and computationally constrained, makes it difficult to perform heavy protection mechanisms. Despite the presence of several solutions for protecting user privacy, they were not created for the purpose of running on small devices at a large scale. On top of that, existing solutions lack the customization of user privacy in which users have little to no control over their own private data. In this regards, we address the aforementioned issue of protecting user's privacy while taking into account efficiency as well as memory usage. The proposed scheme embeds an efficient and lightweight algebra based that targets user privacy and provides efficient policy evaluation. Moreover, an intelligent model to customize user's privacy based on real time behavior is integrated. Experiments conducted on synthetic and real-life scenarios to demonstrate the feasibility and relevance of our proposed framework within IoT environment.

Keywords: Machine Learning, Deep Learning, Access Control, User Privacy, Customized Privacy, Behavior Based Privacy, IoT, SBA-XACML, Limited Resource Devices

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

Chapter 1

Introduction

1.1 Motivations and Problem Statement

Internet of Things (IoT) is attracting a lot of interest from academics and the industry as well due to its high accessibility and diverse set of applications ranging from wearables, sensing as a service, health care, automobiles and smart cities [35] [50]. The nature of IoT devices of being cheap, small, and Internet connected devices allows their integration in various environments for continuous collection of sensitive and fine grained data. Investments in this field are increasing at a rapid rate where, according to [22], expenditure is expected to reach $2 trillion by 2020. Additionally, considering that most applications of IoT are consumer based, and in the era of big data, vendors would try collecting all sort of data about their users in order to make the most out of it. To complicate things even further, vendors would have access to finer and more sensitive data about their users. Unfortunately, users have little to no control over the non-stop collection of their own private data.

IoT, on its own, is not enough to provide a service or achieve an intelligent entity. The main motivation behind this new paradigm is the type of the collected data. As mentioned earlier, the nature of the collected data is fine grained and private. This data would be stored in a data center for analysis and business use. However, it's far from feasible to manually analyze and interpret the data. Thus, different data mining techniques would be used in order to find some meaning of the collected data and perform some action to provide a personalized service for the users. Big data, machine learning and IoT, when combined, provide an ideal personalized service for users. One current research interest in IoT is activity recognition and monitoring. This area has been thoroughly covered by [44], [39], [51] and several others. Different machine learning algorithms are being incorporated in order to recognize user's current activity. Recognizing a user's activity aids in providing a high level context of what the user is currently doing. This context can then be further incorporated with other data to provide a customized service. However, despite all that, researchers are yet to overcome the challenges mentioned by [38]: (1) poor performance results in uncontrolled environment and (2) security and privacy of the collected data.

As mentioned earlier, machine learning is being incorporated in IoT to make the most out of the collected data. Yet, despite all that effort, challenges mentioned by [49] are yet to be adhered. Challenges such as online mobile activity recognition, unsupervised activity recognition and light-weight models that can efficiently run on resource constrained devices are yet to be resolved. Additionally, according to the authors, little to no research has been done to recognize high level activities, which, according to their expectation, is to be one of the next research trend.

On top of that, considering that the IoT paradigm is steadily increasing, the number of deployed devices and the provided services will only increase, leading to more and more collection of data. In order to use these services, users are forced to comply with the vendors terms. This leads to the issue of pervasiveness and intrusiveness of IoT devices. These can be incorporated with ease in an environment without the user notice. This leads to several privacy concerns since these devices continuously collect fine grained data about users. Smart IoT devices, such as speakers, TVs, cameras, smartphones, etc. are being integrated in homes to create a smart home environment. However, according to [7], such devices "leak" privacy information which might reveal sensitive data about the users even though the network traffic is encrypted. Moreover, the authors managed to identify the devices and infer possible behavior such as went sleeping, woke up etc.

Furthermore, the nature of the collected data is private and should be well protected. According to [22], 63% of IoT deployment is consumer based. Thus, protection of privacy for consumers should be of high importance. Additionally, the nature of the devices, being computationally weak aggravates the problem. Therefore, a lightweight and robust solution is required to provide the required protection.

In this regards, several methodologies have been implemented in order to preserve and protect user's privacy based on standard access control techniques such as RBAC and ACL and several others. However, traditional centralized approaches do not scale to the large number of of IoT devices and not suitable to run on resource constrained devices, and thus, new solutions need to be proposed.

Moreover, several solutions have been proposed in order to preserve user's privacy such as differential privacy [21][23][53], K-anonymity [6] and several others. However, these solutions were proposed long before IoT was present and thus, these approaches are used to run on powerful servers, making them inappropriate for computationally weak IoT devices. Other works done in access control such as RBAC, ACL, CapBAC and others [24] [31] [9] solve only part of the problem and fail to give user control over his/her privacy.

1.2 Objectives

The main objective of this thesis is to give users control over their own data and privacy while, at the same time, allowing vendors and services to collect data based on user's behavior. The main objectives are as follows:

1. Address privacy issue in collecting fine grained user data from IoT devices

in which users would have control over their own privacy data.

2. Controlling collection of data based on user's current behavior

1.3 Methodology and Contributions

In this thesis, we aim on tackling the issues mentioned earlier in the objectives. For that, we propose our solution addressing each of the mentioned issues to provide users with customized privacy based on their behavior.

1.3.1 LP-SBA-XACML: Lightweight Algebra based Privacy Preserving Scheme

In order to provide users with customized privacy, we proposed LP-SBA-XACML, a lightweight scheme that builds on top of SBA-XACML [33], relevant for devices with limited resources. Additionally, LP-SBA-XACML gives users complete control over their data, hence deciding what can be shared based on several criteria. Experiment results explore clearly that the proposed platform and constructs perform accurately and efficiently on limited-resource devices.

In this regards, the main contributions in this work are as follows:

- Customized user controlled privacy, where the user is provided with fine-grained control over his/her data

- A semantic based language with dedicated privacy constructs for data collection.

- Lightweight and efficient policy-based evaluation mechanism relevant for IoT devices with very limited re- sources.

1.3.2 Deep Learning-based Approach for Activity Recognition and Privacy Customization

We extend our work from the previous section by linking data collection with user's behavior, in which users would want different data to be collected depending on what they're currently doing i.e. don't access documents if at work. In order to infer user's behavior, we trained a deep neural network model using the TensorFlow framework [3]. Afterwards, we developed formal semantics and integrated the behavior in the evaluation of our proposed LP-SBA-XACML framework. Furthermore, our framework can be used as a behavior based service management, in which users can "subscribe" to a specific service for a certain behavior, and the service would provide some functionality that the user is interested in.

In this regards, the main contributions in this work are as follows:

- A personalized behavioral recognition machine learning model in order to infer user's behavior in real time.
- Formal semantics for LP-SBA-XACML framework.
- Behavior based service management.

1.4 Thesis Organization

The remaining of the thesis is organized as follows: In Chapter Two, a thorough overview about the literature review regarding context aware in IoT and machine learning is presented. In Chapter Three describes our proposed LP-SBA-XACML. First we explain our architecture and how the intelligent model integrates with the language. Second, we present the algorithms for evaluating whether a request or vendor can collect user's data. In Chapter Four we present intelligent behavior detection as well as the performed experiments and performance results. Then we conclude in Chapter Five where we summarize the contributions and future work.

Chapter 2

Background and Related Work

2.1 Introduction

In this chapter, we explain background concepts used in our thesis as well as the literature review in this domain. The following concepts will be covered in the background section: Internet of Things (IoT) and its issues, privacy and its importance, SBA-XACML, machine learning overview and types of algorithms, deep learning and TensorFlow.

2.2 Background

In this section, we represent a brief overview about the concepts of Internet of Things, Security and Privacy concerns, SBA-XACML , Machine Learning and finally TensorFlow framework.

2.2.1 Internet of Things Overview

According to [41], the authors identified four physical elements needed for any IoT device: (1) Sensors for collecting information, (2) Identifiers to identify source of the collected data, (3) Software responsible for processing the data and (4) Internet connectivity. Thus, making IoT "the network of things, with clear element identification, embedded with software intelligence, sensors, and ubiquitous

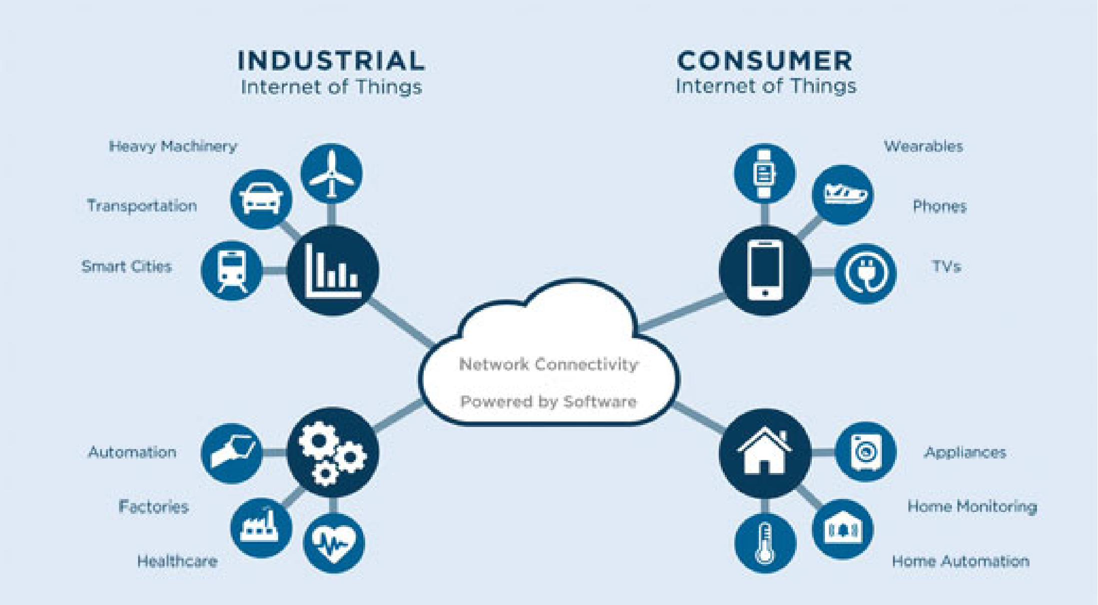


Figure 1: Internet of Things Overview

connectivity to the Internet" . IoT enables everyday "things" and objects to communicate, send and retrieve information, over the Internet. Considering that there will be 20 billion devices by 2020, coupled with the fact that we currently live in the big data age, there will be an exponential explosion of data to analyze and make proper use of to provide customized and personalized services. Figure 1 aids in visualizing how flexible and versatile IoT is and how it can easily connect and find applications in diverse domains.

2.2.2 Information Security and Privacy Concerns

The goal of information security is to protect any information (online, during transmission, on device etc.) from unauthorized access. Information security has several practical real world applications ranging from securing user's personal data up to secret governmental operations. Any secure system must provide the following services: confidentiality, availability, integrity, authenticity and accountability.

- Confidentiality: Preventing unauthorized disclosure to information
- Availability: Information must be available at all times whenever an autho-

rized user wishes to access it.

- Integrity: A secure system should prevent any unauthorized altering of information, thus, preventing "fake" information.
- Authenticity:
- Accountability: Any user performing any action on information must be held accountable for the change. The system must provide a way to verify who did what and when.

Information security is a wide topic with several research paths. For that, we limit the scope in this thesis to personal information generated by users on their personal device and focus on ensuring that user's privacy is maintained. Especially in the digital age, information is being generated at an exponential rate. For that, users need to be able to user different services and generate data while knowing that their personal and private data is safe on their device, especially in the IoT era where most of the data is considered to be private [22].

The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so. [41]. However, this does not mean that IoT does not come without its own set of issues and concerns. The fact that IoT devices are Internet connected devices, this makes them susceptible to security risks such as man in the middle attacks, unauthorized access to device, etc. Furthermore, considering that IoT devices are physical objects, they can actually cause physical damage [16]. To make matters worse, IoT has wide array of applications in health care, which is one of the areas where privacy and security are of top priority. Moreover, considering IoT devices are, in general, computationally weak, new measures need to be taken in order to provide proper security and preserve privacy. Finally, most of the collected data is private, thus, users need their privacy to be maintained [22]. Furthermore, according to [16], attackers can learn about personal life and behavior of users from aggregating metadata of different devices. Thus, information security is of high importance for the IoT era to ensure successful deployment of IoT services

2.2.3 Access Control

Access control is the first layer of security in any system and it plays a critical role in ensuring the security and durability of the system. Where an unauthorized access can lead to serious consequences to any organization. Simply put, access control is the process of deciding whether a user has the right to access or use a certain resource.

There are several models that tackle this issue, however, all models follow the same conceptual flow and try to attain the same goals. As mentioned earlier, the goal of an access control model is to moderate the system and ensure all existing users have the right to access the resources. Figure 2 summarizes the general flow for any access control model where: (1) a user sends his/her credentials to be processed. (2) verify credentials, if the credentials are not valid, then access is instantly denied. Otherwise, (3) load user profile and check privileges. If the privileges enable the user to access the resource/system, grant access, otherwise, access is denied.

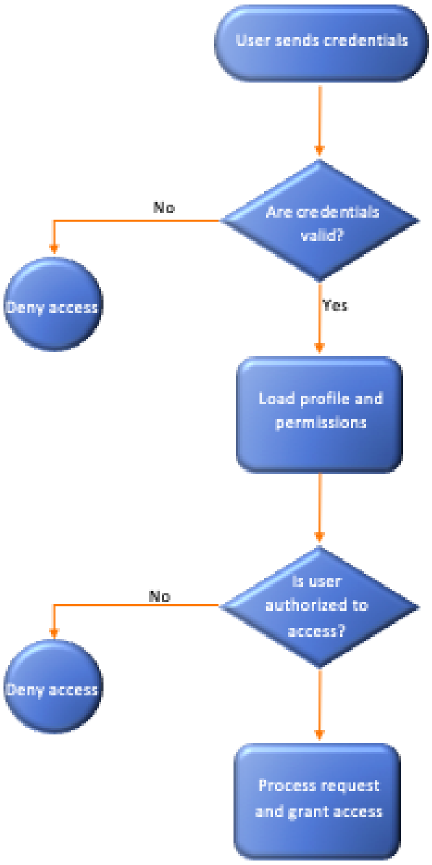

User sends credentials
Are credentials valid?
No
Deny access
Yes
Load profile and permissions
Is user authorized to access?
No
Deny access
Process request and grant access

Figure 2: Access Control Flow

Access control is a huge topic that is continuously changing, where new techniques and concepts are being proposed in the literature to improve existing access control systems. In this thesis, we provide an overview about access control in IoT in the literature review.

2.2.4 XACML

XACML, the eXtensible Access Control Markup Language, is an OASIS standard that is based on XML for creating and enforcing access control policies [33]. XACML is composed of three main components: policy set, policies, and rules. Policy set is a collection of other policy sets or policies. A ”policy combining algorithm”, or PCA, is specified to combine and resolve conflicting policies. A policy, on the other hand, is a collection of one or more rules. Similar to the policy set, a policy must define a ”rule combining algorithm”, or RCA, to resolve

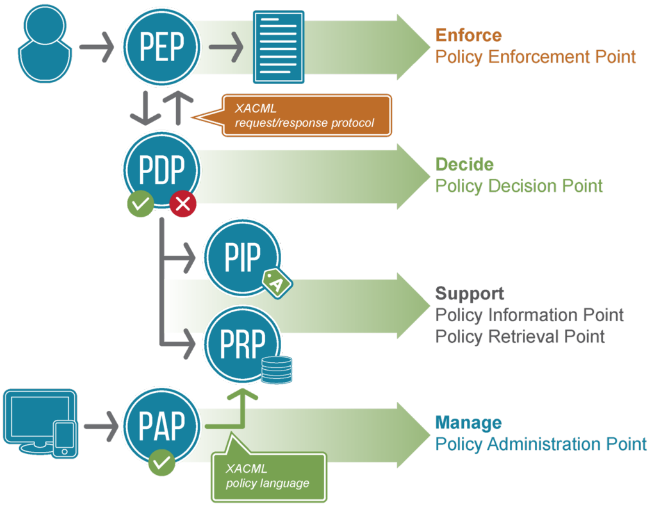


Figure 3: XACML Architecture

conflicting rules. Finally, the rule, which is the smallest component in XACML and it either grants access or denies.

Each of the components has a target, which is composed of zero or more subjects, actions, and resources. If, at any level, the target of a request does not match that of a policy set, policy, or rule, a result of not applicable is returned. Figure 3 displays the architecture for an XACML engine. All XACML policies are maintained by PAP, policy administration point. The PEP, policy enforcement point, is responsible for receiving an XACML request and forwarding it to the PDP, the policy decision point. PDP is responsible for evaluating the request against the policies and reaching a final decision. The PDP might request additional information, such as user profile, and does so by querying the PIP and PRP, policy information/retrieval point.

2.2.5 SBA-XACML

Our work builds on top the work done by [33]. The authors proposed an efficient alternative to the industry standard access control language, XACML, using set algebra. XACML is known to be inefficient in evaluating huge policies [33]. For that, SBA-XACML uses set algebra in order to efficiently reach to a decision. Considering that SBA-XACML is fully compatible and is a lightweight version of

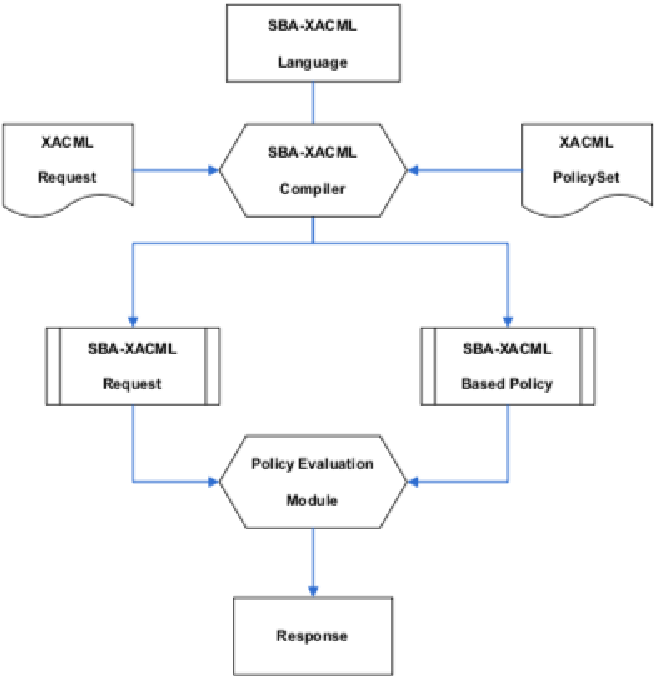


Figure 4: SBA-XACML Architecture

XACML, we build on top of the language and add dedicated privacy constructs.

In this section, we describe an overview of SBA-XACML, how it maps to XACML.

Target

A *target* consists of a set of subjects, resources and actions. It is mapped to SBA-XACML as follows:

$$TR = \{S, R, A\}$$

where S represents a set of subjects, R set of resources, and A set of actions.

Obligations

Obligations can contain one or more obligations, which are methods that execute under a **permit** or **deny** effect. It is mapped in SBA-XACML as follows:

$$OBLs = OBL - SET$$

$$OBL = \{OBLID, FFOn, \{\{AttId, DT, V\}$$

in which *OBL - Set* represents the set of obligation *OBL*, *OBLID* corresponds to the id that uniquely identifies the obligation, *FFOn* is the Fulfill On attribute to determine when the obligation must be enforced (**permit/deny**), *AttID* is the attribute of the obligation to be carried out, DT represents the type of the attribute and V is the attribute value.

Policy Set

A policy set is represented in SBA-XACML as follows:

$$PS ::=< ID, SP, PR, PCA, IPS, OBLs, TR >$$

in which *ID* represents the policy set id, *SP* corresponds to the set of policies that belong to PS, *PR* represents the precedence order of the policies, *PCA* is the policy combining algorithm, *IPS* is the policies or policy set that are referenced by *PS*, *OBLs* is the set of obligations and *TR* is the target.

Policy

A policy is represented in SBA-XACML as follows:

$$P ::=< ID, SR, PR, RCA, OBLs, TR >$$

in which *ID* is the policy id, *SR* corresponds to the set of rules that belong to P, *PR* represents the precedence order of the rules, *RCA* corresponds to rule combining algorithm, *OBLs* is the set of obligations and *TR* represents the target.

Rule

A rule is represented in SBA-XACML as follows:

$$R ::=< ID, RC, TR, RE >$$

in which *ID* uniquely identifies a rule, *RC* corresponds to the set of rule conditions, *TR* is the target and *RE* is the rule effect.

RC is a boolean function that takes a list of parameters be it subjects, resources, actions, or attributes. The *RC* is mapped to SBA-XACML syntax as follows:

$$RC = \{Apply_function, \{parameters\}$$

where $\{Apply_function\}$ represents the function to be called and {parameters} represent the attributes to get passed to the function.

SBA-XACML Request

An XACML request is sent by a subject S to access a specific resource R, and perform some action A. In SBA-XACML, an XACML request is defined as follows:

$$Rq ::= < Sr, Rr, Ar >$$

SBA-XACML Response

An XACML response is the decision after evaluating policies against the request sent. The response is composed of a decision D and obligations *OBLs.* It's mapped to SBA-XACML as follows:

$$Rs ::= < D, OBLs >$$

where D is the final decision after evaluating the policies, and *OBLs* is the set of obligations.

2.2.6 SBA-XACML Architecture

Figure 4 summarizes the modules of SBA-XACML and how policy evaluation is performed. SBA-XACML is comprised of a compiler that takes XACML request and PolicySet as input, and translates them to SBA-XACML language. Once the translation is performed, a module is responsible for evaluating the policy against the received request. Finally, the evaluation engine outputs an SBA-XACML response

2.2.7 Machine Learning Overview

Machine learning falls in one of three categories: (1) Supervised learning, (2) Unsupervised learning, or (3) Semi-supervised learning[15]. According to [1], machine learning can be defined as A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. Simply put, machine learning algorithm is an algorithm that ”learns” from experience. The algorithm updates its parameters based on the data that it received so that it becomes more accurate at predicting and giving better results. In this section, we’ll give a brief overview about different categories in machine learning and about deep learning.

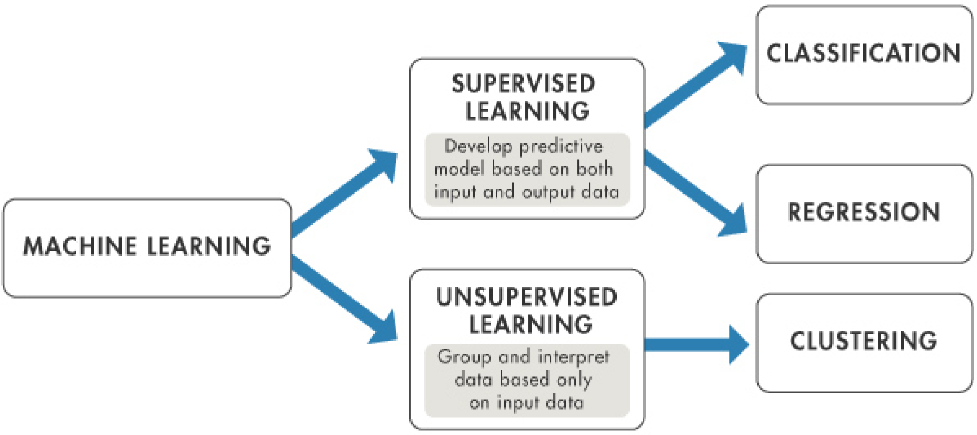

MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Figure 5: Machine Learning Overview

Figure 5 summarizes the main categories of machine learning and the type of prediction or inference that they perform.

Supervised Learning

A machine learning algorithm falls into the supervised learning if it uses labelled data (X and Y) to predict the relation between the data and the result [29] [1]. The algorithm receives the records, outputs some prediction value $\bar{Y}$ and compares it with the actual target value Y. The algorithm then computes an error E to determine how ”far” is the predicted value from the actual target. Once the error is calculated, the algorithm makes some minor ”updates” over its parameters and iterates over the record data again. This process keeps on repeating until the algorithm reaches an acceptable accuracy (set by the developer) or until a certain number of iterations is reached. It is important to note that the type of data that you have determines whether or not you can use a supervised approach.

In Figure 5, there are two types of predictions supervised learning algorithm can perform: (1) Classification or (2) Regression. Classification is when we have categorical data, that is, data that belongs to a finite set of classes. An example of categorical data is image classification. Given an image, detect if its an image of a flower, person etc. Theres a limited set of categories an image can belong to. Regression, on the other hand, is when the output is continuous. A simple example of regression is predicting stock price, where price is a continuous value.

Unsupervised Learning

In unsupervised learning, same as before, we have data record X, except that we are missing the target value Y. This means that our data is unlabelled. So we do not have a ”reference” or ”guide” for our algorithm since an error E cannot be computed [29] [1]. Usually, the goal in unsupervised learning is to ”learn” some relation or discover some pattern in the data. More specifically, we want to learn the probability distribution of the dataset [1]. Usually, clustering algorithms are used in such scenarios since they ”group” or ”cluster” data such that records in the same ”group” or ”cluster” have the same pattern or belong to the same category as shown in Figure 5

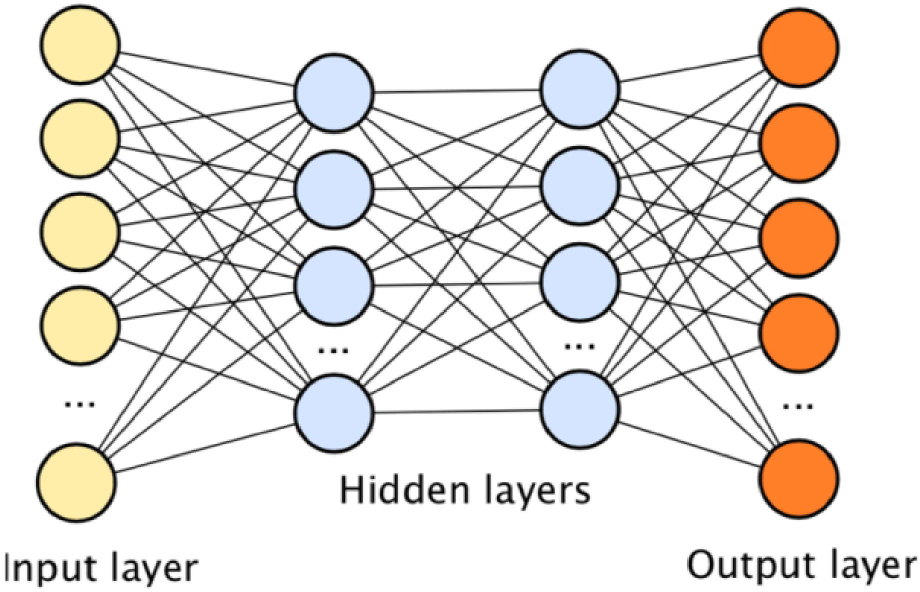


Figure 6: Deep Neural Network

Semi-supervised Learning

This type of learning is when we have data, and only part of the data is labelled. What happens is that the algorithm lears from the set of labeled data and tries predicting or inferring the values of the unlabelled ones [15] [29].

Deep Learning

Deep learning uses a technique called Deep Neural Network (DNN). DNNs have achieved high level of accuracy in computer vision, translation, robotoics, games and speech recognition [15]. DNN is a type of Neural Network (NN), where its simply a bunch of nodes in a layer, and these nodes are connected to nodes in other layer. A NN is called ”deep” because it has multiple ”layers” of neurons [15] [1]. Usually, a DNN requires lots of data in order to be trained properly and is computationally demanding. However, if trained properly, DNN models achieve best results. Figure 6 helps give a visual representation of a deep neural network, where the first layer is our input layer, the last layer is our output layer, where the network actually gives the output, and all the layers in between are called hidden layers. We can have as many hidden layers as we want, there’s no

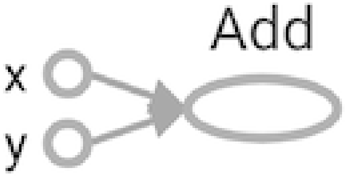


Figure 7: A Simple Flow Graph

"rule" to set the number of hidden layers and number of neurons in each hidden layer. However, it's important to keep in mind that the more layers we have, the more complex the model becomes and the more time and computational power needed to train the model.

TensorFlow

TensorFlow is a machine learning framework created by Google that enables developers to create models that run on wide array of devices and platforms such as Android, iOS, raspberry pi, Linux, Windows and macOS [3]. TensorFlow uses dataflow graph to represent computation dependency between operations. Nodes in the graph represent a unit of computation or operation and the edges represent the "flow" of "tensors" (data record) from one node to another in a directed manner. Figure 7 shows is a simple flow graph that adds two variables. The variables, x and y, are defined as "tensors", and the operation "Add" is represented as a node in the graph. Since the operation is performed on the variables, the x and y tensors are "connected" to the operation. It's important to note that the connection is "directed", that is, the connection is only from x to the Add operation, and from y to the Add operation. So when traversing the graph, we cannot go from Add to any of the tensor variables.

2.3 Related Work

Our approach is based on customized and fine-grained privacy preserving access. As such, we provide in this section the related work to the following categories: (1)

privacy customization and access control, (2) activity and behavior recognition

2.3.1 Privacy Customization and Access Control

Privacy Preserving Techniques

Authors in [6] discussed some common privacy preserving techniques: randomization method and k-anonymity with l-diversity. The randomization method adds noise to the data in order to mask the attribute values of records such that individual record values cannot be recovered. This is the only one computationally acceptable that can be done during data collection, without the need for a trusted server. However, one downside of this approach is treating all records of equal weight, making outlier records difficult to mask. Additionally, high dimensional data reduces the effectiveness of this method and makes it susceptible to adversarial attacks [5]. An alternative to the randomization method is anonymizing data. A well known technique is k-anonymity and l-diversity. This approach requires that ”every tuple in the table be indistinguishable related to no fewer than k respondents”. It is important to note that optimal anonymization has been shown to be NP-hard. Therefore, researchers tend to use approximation algorithms that provide a guarantee on the quality of the solution to be within a certain factor. Additionally, k-anonymity is susceptible to homogeneity and background knowledge attacks. Therefore, it’s rarely used by its own as means to anonymize data, and is commonly integrated with l-diversity.

Other privacy preserving techniques mentioned by [36] are secure multi-party computation (SMC) and homomorphic encryption (HE). SMC is a subfield of cryptography and aims at creating methods for two or more parties to cooperate on computing a function based on private inputs. It assumes that each party is willing to contribute some data, however, the party contributes privately, without letting other parties know the shared input. Despite accomplishing its goals, SMC is known to be inefficient and is simply not feasible on IoT devices. HE, on the other hand, is a form of encryption that allows computations on ciphertexts that generate an encrypted result such that when decrypted, it matches the result as if it was performed on plaintext. The purpose of HE is to perform operations on encrypted text without affecting privacy, confidentiality and integrity of the message. Unfortunately, similar to SMC, HE is computationally intensive and cannot be performed on a limited resource devices.

Authors in [11] proposed a framework for an efficient energy management for preserving user privacy in smart grid environment. The proposed architecture constitutes of IoT devices, IoT gateways, which are considered to be computationally powerful, fog devices and the cloud. Upon collecting energy data for each user, the data is sent to the fog devices and are aggregated. Once aggregated, you can no longer identify individual user data. Unfortunately, the authors did not implement the framework and was left as future work.

Finally, differential privacy (DP), used by the industry giants Apple and Google [20], is another technique that aims at preserving privacy of users. DP is mostly used in database security and data mining where a database is made publicly accessible. However, individual identity should not be exposed. One common technique is to anonymize data. Unfortunately, data anonymization is not always feasable shown in [34]. The authors have won the Kaggle Social Network challenge by de-anonymizing the data with 90% accuracy by introducing a simulated annealing-based weighted graph matching algorithm. For that, DP is the de-facto standard when it comes to sharing public data privately. DP ensures the outcome of a calculation to be insensitive to any particular record in the dataset [53]. Several publications propose DP as a means to collect data and apply data mining algorithms while maintaining user privacy [21], [23] and [53]. However, there is a tradeoff between privacy and accuracy. Generally speaking, the more privacy you want for the users, the less accurate the results are going to be, and vice versa. Therefore, you need to strike a balance between privacy and accuracy. Unfortunately, DP has two main drawbacks: (1) it requires a lot of computational power, which is not feasible on IoT devices, and (2) most of the work done in DP assume that the data collector is trustworthy.

Privacy Related and Dynamic Access Control

Traditional access control methods such as RBAC and ACL are not fit for IoT due to its decentralized architecture, heterogeneity, and high scalability. To overcome such limitations, [24] proposed a capability based access control (CapBAC). A capability is a communicable, unforgeable token of authority. It refers to a value that uniquely references an object along with an associated set of access rights. Once a user has a capability, s/he can access the object as specified by the generated capability.

[31] proposed the usage of elliptic curve cryptography (ECC) for being a relatively lightweight public key cryptography compared to RSA and symmetric key cryptography (SKC). Additionally, they proposed the usage of OpenID technology that allows users to have a single account and can have access to different sites without the need to authenticate at each one. Finally, they incorporated RBAC in order to enforce access control. The authors claim that there approach is safe against eavesdropping, man-in-the-middle, replay, and key control attacks. Unfortunately, no experiments where conducted to prove such claims. Additionally, RBAC, as mentioned earlier, is simply not suitable for IoT environment due to scalability issues.

[28] provided the first fully implemented two-way authentication security scheme of DTLS on IoT, using RSA as their PKC algorithm. The authors managed to perform a fully authenticated DTLS handshake based on an exchange of X.509 certificates, using 2048-bit RSA keys. The experiments where performed on real IoT devices, with a memory limit of 48 KB RAM, with their implementation consuming less than 20 KB RAM. The provided implementation provides message integrity, confidentiality and authenticity.

One important criteria to note when it comes to access control, as mentioned in [36], if an access control methodology requires a user to expose attributes in order to be granted access, then privacy violation problems can arise. It is necessary to maintain a balance between access control and privacy without undermining one or the other.

[9] proposed a dynamic risk-based access control that uses real time environment features, such as location and time, to reach an access decision. A security risk value is calculated based on the sensitivity of the data and type of operation to be performed for each access request and is then compared with predefined risk policies to reach a decision. In order for the approach to be dynamic, the proposed model monitors user's behavior for anomaly detection. The monitored user behavior is compared with smart contract, which is a software code that runs on blockchain, to ensure that the user acts as expected.

Authors in [26] proposed a user interactive privacy preserved access control in IoT. They proposed a Human Interactive Security Protocol (HISP), which is a protocol that enables users to publish IoT data in different levels of security. Additionally, they proposed context aware k-anonymity for preserving privacy. Depending on the context, access control rules or policies are generated. One downside of their approach is lack of experimentation and performance tests. Additionally, k-anonymity is known to be NP-hard [6], which makes it infeasible to run on IoT devices.

Authors in [10] proposed a context aware usage control for the web of things. Web of Things (WoT), as described by the authors, encapsulates functionalities from IoT into publishable services on the web, hence providing a seamless integration between IoT and web. IoT devices connect to the "Device abstraction layer", which is a layer that allows communication between different network protocols. Then the services are provided by the "Web Service Provider", which is basically an application server that is connected to the Internet. Finally, they used XACML in order to control access on the IoT devices.

A dynamic risk-aware access control model with XACML implementation for the IoT platform was proposed by [40]. The authors applied the association rule learning (ARL) algorithm in order to calculate the risk of accessing an object. The algorithm takes into account the history of the user in order to calculate the risk value.

Finally, work done by [8] proposed an adaptive risk-based access control for the IoT system. Access is granted if the calculated risk value is below a certain threshold. The risk is calculated based on user history and contextual information e.g. time and location. Finally, user behavior is continuously monitored in the system, in real time, through the use of smart contracts. The risk value is continuously monitored and re-evaluated, so that whenever it crosses the threshold, even when access is granted, the user is no longer allowed to access the resource.

To the best of our knowledge, non of the approaches so far have integrated machine learning in access control in order to intelligently provide a dynamic access control system based on user's behavior.

2.3.2 Activity and Behavior Recognition

Simple Activity Recognition

Simple activities, as defined by [44], are activities that are repetitive by nature and can be easily recognized using one sensor such as accelerometer.

Authors in [39] managed to recognize eating gestures through the use of wrist-worn sensors (accelerometer and gyroscope) using the temporal sequencing technique. The authors compared Hidden Markov Model (HMM) and K-Nearest Neighbor (KNN) to identify four different eating gestures: utensiling, bite, drink, and rest. HMM proved to better recognize gestures with 84.1% accuracy compared to 71.7% for KNN.

Work done by [51] developed an Android application for data collection and created a universal model for activity recognition. The following activities were recognized: walking, sitting, stairs, jogging and standing. The authors created a universal model, which achieved an average of 75% accuracy. Once a user is actively using the application, a personalized model is developed, achieving 95%

accuracy.

A window based algorithm was proposed by [47] using the Support Vector Machine (SVM) to detect activities and motions within an activity. The authors achieved 91% accuracy for simple activities, however, for motion detection, accuracy was much lower at 80%. According to the authors, such results are understandable due to the short time window to recognize the motion.

Detecting activities using sensors is far from a simple task, especially when their are myriad of sensors in a smartphone and the various positions that the device is held. For this reason, authors in [45] identified which key sensors (and their combination) play an integral role for activity recognition, as well as measuring performance in four different positions. The authors concluded that the combination of accelerometer and gyroscope yield, in most scenarios, better results than being used separately. Additionally, the accelerometer performs, in general, better than the gyroscope to detect the performed activity.

Complex Activity Recognition

Simple and complex activities were identified by [44], in addition to analyzing the different window size (2-30 seconds) and their efficiency in recognizing complex activities. The identified complex activities are: upstairs, downstairs, coffee, talk, smoke and eat. The authors concluded that larger window size resulted in better capturing complex activities.

Authors in [14] recognized complex activities performed in a smart home environment. Simple activities such as walking, sitting etc. were detected using smartphone accelerometer. Room detection using BLE beacons. Finally, accelerometer from the smartwatch combined with the activity and room are used in order to detect complex activities such as sweeping, vacuuming, writing, reading etc. Another critical work done by [48] was defining complex fine-grained activities in a smart home environment. The authors managed to identify 22 such activities including cooking, sit and eat, lying on bed etc.

Context Activity Recognition

Work done by [32] enhance the display of information depending on the context i.e. user activity and location. The authors identified three activities: walking, running and standing using the SVM model. If the user is walking or running, then reduce the amount of information displayed on the device and increase font size in order to provide enhanced user experience. Additionally, they used location context when the user is standing to display relevant information such as nearby restaurants.

Authors in [12] took a different approach, in which they studied how context, be it physical or social, can affect the behavior of individuals and assist in the formation of healthy habits. The authors focused on the health aspect, stating that wearable sensors can aid in inferring, in real-time, user behavior and take necessary action when mixed with physical and social context.

Authors in [52] proposed a machine learning, context-aware system that provides a service depending on the current context in a smart home environment. Finally, the concept of adaptive context-aware clinic for heart failures was proposed by [4]. For that, they implemented an SVM with RBF kernel, achieving an overall accuracy of 82.0%.

Intelligent Behavior Recognition

Machine learning has diverse set of applications when it comes to behavior recognition from predicting transportation mode (walking—cycling—driving—public transportation) [37] to a personalized recommendation framework based on user's clicks, queries and history[55] and [13].

Authors in [42] proposed an improved Naive Base classifier in order to predict user's phone call behavior(accepting, rejecting, missed, or outgoing). Similarly, work done by [43] proposed a rule association based solution to extract a concise set of behaviors.

A different approach was taken by [54], where they used tapping behavior of the user for authentication. The authors capture the user's tapping speed, pressure and time in order to authenticate, achieving an error rate as low as 3.65%.

Authors in [27] described an open framework for detecting and capturing user behavior from smart meter power consumption data using the SVM model. They managed to attain an accuracy of 94%. They used machine learning techniques in order to capture events performed by the user, obtaining a 94% accuracy using SVM model. Hidden Markov Model (HMM) is a very common model for predicting user behavior and the next action to be taken. For this reason, authors in [52] proposed an improved HMM to support people with disabilities in a smart home environment. The authors limited the scope of their work for regulating temperature since this is a very common task and can be very irritating for people with disabilities. An optimal accuracy of 78% was achieved, compared to only 65% using the traditional HMM, a 13% improvement.

2.3.3 Analysis

Privacy, especially when it comes to IoT, is still an open problem with no one size fits all. Additionally, to the best of our knowledge, a customized privacy solution is yet to be provided. When it comes to activity recognition, most of the conducted work focuses on recognizing simple activities. Few researchers started to branch out and recognize application specific activities e.g. sub steps when it comes to eating, activities in basketball etc. However, the literature still lacks a personalized, general high level activity recognition.

For that, our work differs in three aspects: (1) The user has complete control over his/her data, where processing is performed locally on the device. (2) an efficient and memory friendly solution for resource constrained devices. (3) personalized machine learning model where inference runs in real time, offline and locally on the device.

2.4 Conclusion

In this chapter, we briefly covered the background and literature review. We've covered basic concepts related to IoT, Security and Privacy, SBA-XACML and Machine Learning. Finally we summarized the literature review related to our research and showed existing limitations.

Chapter 3

LP-SBA-XACML: Lightweight Algebra based Privacy Preserving Scheme

3.1 Introduction

We are experiencing an exponential increase in the amount of user generated data. According to [46], 5 quintillion bytes of data are being produced every day. Furthermore, they estimate that an entire lifetime is needed to manually analyze the generated data of a single sensor. The sheer amount of data is huge, and theres an urgent need to automate the process and make practical use out of it. For that, data mining and machine learning tools are the best techniques so far to deal with this problem. However, we must keep in mind that most of this data is user generated, thus, it's of high importance to protect the data and user's privacy.

In this chapter, we will present our work to give users some control over their data and preserve their privacy. Section 3.2 presents an overview of our proposed approach and architecture. Section 3.3 describes the language and the privacy constructs. Afterwards we explain the formal semantics in Section 3.4. Finally, performance tests are presented in Section 3.5.

3.2 Approach Overview and Architecture

Our approach makes use of the industry standard XACML. However, it's known to have less than ideal performance. Despite several work done to tackle the issue such as [18], [19], [17] and [30], they propose a major structural change in the standard [33]. For that, we decided to go build on top of SBA-XACML due to its compatibility with XACML and it's fast performance.

Figure 8 illustrates the architecture of our proposed approach, which is composed of three main modules: (1) Set of entities requesting access to sense data, (2) IoT devices responsible of sensing and collecting data, and (3), LP-SBA-XACML platform running on IoT devices to preserve the privacy of collected data.

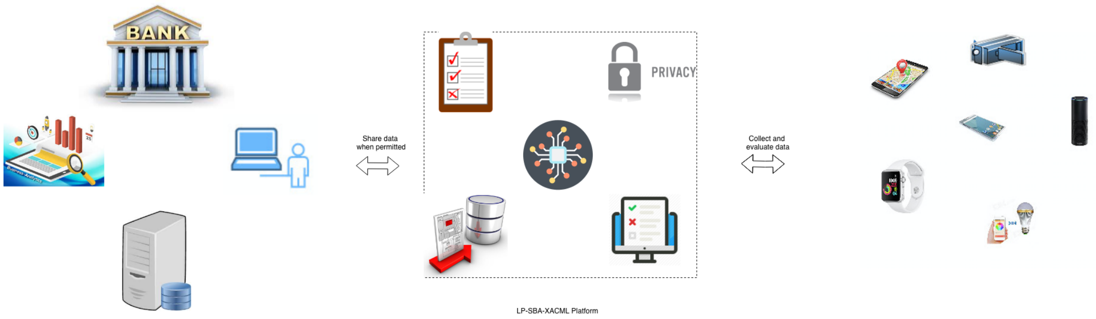


Figure 8: Scheme Architecture

Below is a detailed description of our proposed architecture:

1. IoT devices data collectors: Small, internet connected devices that are responsible to sense and collect sensitive data about the user and his/her surroundings.

2. User/system application: An entity that is interested in collecting the data from IoT devices for statistical analysis and applications.

3. LP-SBA-XACML Platform: This is the core module of our approach composed of the following components:

(a) LP-SBA-XACML language: is a semantic-based privacy-preserving language built on top of SBA-XACML [33], which is an efficient and compatible alternative for XACML with lighter policy evaluation mechanisms. Detailed description of the newly proposed constructs is provided in Section 3.3. The privacy constructs enable fine-grained to user data based on several factors and user's behavior.

(b) Policy Evaluation module: is responsible of loading the appropriate policy in order to assess their content with respect to the request components and provide the type of access decision and preserve user's privacy.

(c) Behavior Detection Module for Privacy Enforcement: This is tightly integrated with the language since the evaluation is behavior oriented. That is, the result is tightly coupled to user's behavior. The behavior module is responsible to infer user's current behavior through the use of a deep neural network implemented using the TensorFlow framework [3]. The appropriate action is taken based on the predicted behavior. We explain in more details about the module in Section 4.2

3.3 LP-SBA-XACML Language Description

3.3.1 LP-SBA-XACML Language Description

As mentioned earlier, LP-SBA-XACML builds on top of the SBA-XACML language. In this section, we introduce the newly added constructs and some changes in evaluation semantics to accommodate them. Moreover, we introduce a minor change in the mapping of policy set, policy, and rule to LP-SBA-XACML in order to accommodate user's behavior.

New Privacy Construct

A privacy construct PC is defined as follows:

$$PC ::=< L, N, T, BC >$$

where L is a set of location constraints, N is a set of network constraints, T is a set of time constraints, and BC is a set of defined behaviors.

More formally, a location L is defined as follows:

$$L = \{LatLng_1, LatLng_2, ...LatLng_{\mathrm{n}}\}$$

Similarly, a network N is defined as follows:

$$N = \{(Type, IP - Range)_1, (Type, IP - Range)_2...(Type, IP - Range)_{\mathrm{n}}\}$$

A time T is defined as follows:

$$T = \{(Date - Range, Time - Range)_1, (Date - Range, Time - Range)_2...(Date - Range, Time - Range)_{\mathrm{n}}\}$$

Finally, BC is defined as follows:

$$\{B_1, B_2...B_{\mathrm{n}}\}$$

3.3.2 Policy Set

A policy set in LP-SBA-XACML is defined as follows:

$$PS ::=< ID, SP, PR, PCA, IPS, OBLs, PCs, TR >$$

The only change is the introduction of the PCs premise. Where PCs is simply a set of privacy constructs.

3.3.3 Policy

Similarly, a minor change has to be made at the policy level:

$$P ::=< ID, SR, PR, RCA, OBLs, PCs, TR >$$

Where a policy P can have one or more privacy constructs, thus, a set of constructs, **PCs**, is passed as parameter

3.3.4 Rule

Finally, we introduce the mapping of the rule in LP-SBA-XACML:

$$R ::=< ID, RC, PC, TR, RE >$$

Where a rule R takes a single privacy construct.

3.3.5 New Anonymous Communication Mechanisms

As mentioned earlier, our work is to preserve and customize user's privacy. For that, appropriate measures need to be taken to protect user's privacy especially between device communication. Thus, we implemented the approach proposed by [25], a TOR based anonymous communication to secure smart home appliances. TOR was installed and configured at the access point, so that every request coming out of the smart home appliances will be rerouted and passed through TOR. In our approach, TOR was installed on a smart device, and requests coming from the IoT device would get rerouted to the smart device, and the request would be sent by TOR. That way, we ensure private communication without exposing the user's identity.

The approach has been implemented as an obligation. If the user wishes to send a request anonymously, then an obligation that does so is added. Below is an example that demonstrates how it works:

$$P ::=< \\ P1, R1, R1, permit - overrides, \{anonymousObligation, permit, \{send - anonymously, , \} >$$

where we have a policy with **PCA** of *permit-overrides*, and an obligation with **OBLID** set to *anonymousObligation*, with **FFOn** *permit*, and **AttId** set to *send-anonymously* with empty DT and V. If the decision of the policy is permit, then the obligation is fired and the method *send-anonymously* will take care of rerouting the response through TOR.

Moreover, we are currently working on developing more functions to provide the user with alternatives for anonymous communication based on the related work in the future.

3.4 LP-SBA-XACML Formal Semantics

In this section, we represent the matching semantics for the newly proposed privacy construct PC, as well as the newly proposed evaluation semantics for each of the policy set, policy, and rule affected by them. Recall that the PC contains a set of contextual constraints, defined by the user, that must be met. For that, the PC is evaluated against the user's device in real time. That is, contextual data, such as time and location, will be retrieved whenever a request is received.

First, we'll define the matching semantics for each of the location, network, and time premises accordingly.

Rules 1 and 2 represent the matching semantics for a location constraint L. First, we check if there exists a location ℓ in the set defined by the user that is equal to the current user location. If such an ℓ exists, then the location matches to True. If, on the other hand, no such ℓ exists, then the location is matched to False.

Rules 3 and 4 represent the match semantics for a network constraint n. A network matches to true if there exists an n such that the network type equals to

Table 1: Matching Semantics for Location

$$\frac{\exists \ell \in L; \ell = UL}{< L, UL > \underset{match}{\vdash} True} \qquad \textbf{(Rule 1)}$$

$$\frac{\forall \ell \in L; \ell \neq UL}{< L, UL > \underset{match}{\vdash} False} \qquad \textbf{(Rule 2)}$$

the user's current network type i.e. work, home, cafe etc., and, the user's current IP belongs the set of IP-Range. If, on the other hand, no such n exists, then the network matches to False (Rule 4). x

Table 2: Matching Semantics for Network

$$\frac{\exists n \in N; n.\mathit{Type} = UN.\mathit{Type} \wedge \exists IP \in n.IP\_\mathit{Range}; UN.IP = IP}{< N, UN > \underset{match}{\vdash} True} \qquad \textbf{(Rule 3)}$$

$$\frac{\forall n \in N; UN.\mathit{Type} \neq n.\mathit{Type} \vee \forall IP \in n.IP\_\mathit{Range}; UN.IP \neq IP}{< N, UN > \underset{match}{\vdash} False} \qquad \textbf{(Rule 4)}$$

Finally, Rules 5 and 6 represent the time match semantics. If there exists a time t such that the current time $\in t$ time range and the current date $\in t$ date range, then it matches to True. Otherwise, if no such t exists, then the time matches to False.

Once we've defined the matching semantics for each of the privacy premises, we're ready to define the semantics for the privacy construct PC. Rules 9 and 10 represent the matching semantics. If all of the premises, L, T, and N match to true, then the PC matches to true as well. If, on the other hand, any of the premises does not match to true, then PC matches to false

Table 3: Matching Semantics for Time

$$\frac{\exists t \in T; UT.D \in t.DR \wedge \exists T \in t.TR; UT.T = T}{< T, UT > \underset{match}{\vdash} True} \quad \textbf{(Rule 5)}$$

$$\frac{\forall t \in T; UT.D \notin t.DR \vee \forall T \in t.TR; UT.T \neq T}{< T, UT > \underset{match}{\vdash} False} \quad \textbf{(Rule 6)}$$

Table 4: Evaluation Semantics for PRIV_CONST

$$\frac{(L \underset{match}{\vdash} True) \wedge (N \underset{match}{\vdash} True) \wedge (T \underset{match}{\vdash} True) \wedge (BC \underset{match}{\vdash} True)}{< PRIV\_CONS, R > \underset{eval}{\longrightarrow} PermitObl} \quad \textbf{(Rule 9)}$$

$$\frac{(L \underset{match}{\vdash} False) \vee (N \underset{match}{\vdash} False) \vee (T \underset{match}{\vdash} False) \vee (BC \underset{match}{\vdash} False)}{< PRIV\_CONS, R > \underset{eval}{\longrightarrow} Deny} \quad \textbf{(Rule 10)}$$

3.4.1 Rule Evaluation

The evaluation semantics for the rule in LP-SBA-XACML for a request Rq are presented in this section.

The rules 11, 12, and 13 in table 5 represent the evaluation semantics of a rule at the policy level. A rule R is evaluated to permit (Rule 11) if the target matches against the request, rule condition is true, and the privacy construct PC evaluates to permit. On the other hand, a rule evaluates to deny (Rule 12) if the same conditions hold, having the rule effect to Deny. Finally, a rule is evaluated to NotApplicable (Rule 13) if the target does not match, or if the rule condition is false, or if the privacy construct evaluates to deny.

[h]

Table 5: Evaluation Semantics of a Policy Rule

$$\frac{(< TR, Rq > \underset{match}{\vdash} True) \wedge (RC = True) \wedge (PC \underset{eval}{\longrightarrow} PermitObl) \wedge (RE = Permit)}{< R, Rq > \underset{eval}{\longrightarrow} Permit} \quad \textbf{(Rule 11)}$$

$$\frac{(< TR, Rq > \underset{match}{\vdash} True) \wedge (RC = True) \wedge (PC \underset{eval}{\longrightarrow} PermitObl) \wedge (RE = Deny)}{< R, Rq > \underset{eval}{\longrightarrow} Deny} \quad \textbf{(Rule 12)}$$

$$\frac{(< TR, Rq > \underset{match}{\vdash} False) \vee (RC = False) \vee (PC \underset{eval}{\longrightarrow} Deny)}{< R, Rq > \underset{eval}{\longrightarrow} NotApplicable} \quad \textbf{(Rule 13)}$$

3.4.2 Policy Evaluation Semantics

The evaluation semantics for the policy in LP-SBA-XACML for a request Rq are presented in this section.

Rule 14, 15 and 16 presented in table 6 show the evaluation semantics of a policy with a rule combining algorithm (RCA) set to permit-overrides. Note that for RCA deny-overrides, the same semantics apply.

A policy is evaluated to permit (Rule 14) if the target matches, and if there exists a rule that evaluates to permit, and if every privacy construct evaluates to permit as well. On the other hand, a policy is evaluated to deny (Rule 15) if the target matches, if all of the rules evaluates to deny, and all the privacy constructs evaluate to permit. Finally, a policy evaluates to NotApplicable if the target does not match, or there exists a rule that evaluates to NotApplicable, or if there exists a privacy construct that evaluates to deny.

3.4.3 Policy Set Evaluation

The policy set evaluation semantics against a request Rq are represented in this section.

Rules 17, 18 and 19, presented in table 7 illustrate the evaluation for a policy set with policy combining algorithm (PCA) set to Permit-Overrides. A policy

Table 6: Evaluation Semantics of a Policy where (RCA=Permit-Overrides)

$$\frac{\begin{array}{c}(RCA = Permit - Overrides) \quad \wedge \\ (< TR, Rq > \underset{match}{\vdash} True) \wedge (\exists R \in SR; < R, Rq > \underset{eval}{\longrightarrow} Permit) \wedge (\forall pc \in PCs; pc \underset{eval}{\longrightarrow} PermitObl)\end{array}}{< P, Rq > \underset{eval}{\longrightarrow} Permit, OBLs} \quad \textbf{(Rule 14)}$$

$$\frac{\begin{array}{c}(RCA = Permit - Overrides) \quad \wedge \\ (< TR, Rq > \underset{match}{\vdash} True) \wedge (\forall R \in SR; < R, Rq > \underset{eval}{\longrightarrow} Deny) \wedge (\forall pc \in PCs; pc \underset{eval}{\longrightarrow} PermitObl)\end{array}}{< P, Rq > \underset{eval}{\longrightarrow} Deny, OBLs} \quad \textbf{(Rule 15)}$$

$$\frac{\begin{array}{c}(RCA = Permit - Overrides) \quad \wedge \\ ((< TR, Rq > \underset{match}{\vdash} False) \vee (\forall R \in SR; < R, Rq > \underset{eval}{\longrightarrow} NotApplicable)) \vee (\exists pc \in PCs; pc \underset{eval}{\longrightarrow} Deny)\end{array}}{< P, Rq > \underset{eval}{\longrightarrow} NotApplicable} \quad \textbf{(Rule 16)}$$

set evaluates to permit (Rule 17) if the target matches, there exists a policy that evaluates to permit, and all of the privacy constructs evaluate to permit as well. On the other hand, a policy set evaluates to deny if the target matches, all of the policies evaluate to deny, and all of the privacy constructs evaluate to permit. Finally, a policy set evaluates to NotApplicable if the target does not match, or there exists a policy where it evaluates to NotApplicable, or there exists a privacy construct that evaluates to deny.

Table 7: Evaluation Semantics of a PolicySet where (PCA=Permit-Overrides)

$$\frac{\begin{array}{c}(PCA = Permit - Overrides) \wedge \\ (< TR, Rq > \underset{match}{\vdash} True) \wedge (\exists P \in SP; < P, Rq > \underset{eval}{\longrightarrow} Permit) \wedge (\forall pc \in PCs; pc \underset{eval}{\longrightarrow} PermitObl))\end{array}}{< PS, Rq > \underset{eval}{\longrightarrow} Permit, OBLs} \quad \textbf{(Rule 17)}$$

$$\frac{\begin{array}{c}(PCA = Permit - Overrides) \wedge \\ (< TR, Rq > \underset{match}{\vdash} True) \wedge (\forall P \in SP; < P, Rq > \underset{eval}{\longrightarrow} Deny) \wedge (\forall pc \in PCs; pc \underset{eval}{\longrightarrow} PermitObl)\end{array}}{< PS, Rq > \underset{eval}{\longrightarrow} Deny, OBLs} \quad \textbf{(Rule 18)}$$

$$\frac{\begin{array}{c}(PCA = Permit - Overrides) \wedge \\ ((< TR, Rq > \underset{match}{\vdash} False) \vee (\forall P \in SP; < P, Rq > \underset{eval}{\longrightarrow} NotApplicable)) \vee (\exists pc \in PCs; pc \underset{eval}{\longrightarrow} Deny)\end{array}}{< PS, Rq > \underset{eval}{\longrightarrow} NotApplicable} \quad \textbf{(Rule 19)}$$

3.5 Performance Evaluation

In this section, we explore the performance tests in terms of time and memory usage for LP-SBA-XACML with respect to the standard policy-based approach XACML. The experiments were conducted on two devices: Macbook Air, 2.2GHz Intel core i7 with 8GB of RAM and Galaxy S3 with 1GB of RAM.

In order to test XACML, we used the open source Balana implementation [2], which builds on top of Sun's implementation and adds support to the latest version of XACML 3. Additionally, we've ported the library on Android in order to run it on S3. It's important to note that not all of the methods we've used in LP-SBA-XACML are already existing in XACML. In this regards, we have developed the corresponding functions and integrated them in the ported XACML platform.

The experiments were conducted on real-world and synthetic policies in order to show performance on small and large scale policies. Synethetic policy sets are created such that every policy and rule in the set needs to be evaluated in order to reach a final decision. The size of policy sets ranges from 100 to 4000 rules in order to show the scalability and performance on varying policy sizes, where rules are evenly split over the policies. In order to force an exhaustive evaluation of the policy set, we specified a (1) policy combining algorithm *Deny-Overrides*, (2) a rule combining algorithm *Deny-Overrides* has been set for each policy, (3) every rule in the policy has a permit of *Deny* and finally (4) a non empty target is set. Comparisons have been made between XACML and P-SBA-XACML in terms of time and memory usage, where each test was performed ten times to get the average.

Figure 9 summarizes the execution time of XACML and LP-SBA-XACML on desktop, and shows that our approach is orders of magnitude faster than XACML. LP-SBA-XACML is 3.5-13 times faster compared to XACML, depending on the policy size. Figure 10 summarizes execution time on mobile S3. Again, LP-SBA-XACML is consistently faster. LP-SBA-XACML is 2-4 times faster depending on the size of the policy. Figure 11 summarizes memory consumption of both approaches where LP-SBA-XACML consumes 2-22 times less memory.

Moreover, table 8 summarizes the execution time of real world policies run on desktop. The obtained results show our approach to be 7.5-30 times faster, depending on the size of the policies. Table 9 shows the results on mobile, where our approach is 12-28 times faster than XACML. Finally, we we measured memory consumption of real life policies shown in table 10. The obtained results show the memory consumption on mobile where LP-SBA-XACML consumes 44.8-48.8 times less memory. When it comes to memory consumption on Android, you should take into consideration that an application is sandboxed and is given a limited amount of memory that it can consume. And if an application tries to exceed that limit, either the garbage collector (GC) gets called to free some memory, or the app fails. Frequent GC calls lead to higher CPU and energy usage, leading to a faster deplete in battery. In our case, when testing XACML on S3, GC was frequently called regardless of the policy size in order to free memory. This shows serious memory limitations when it comes to the usage of XACML on a resource constrained devices.

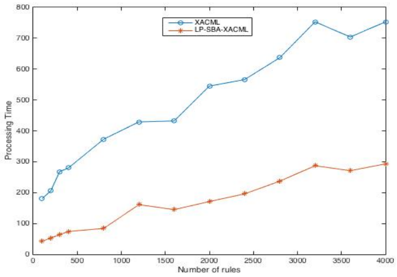


Figure 9: Performance results on desktop

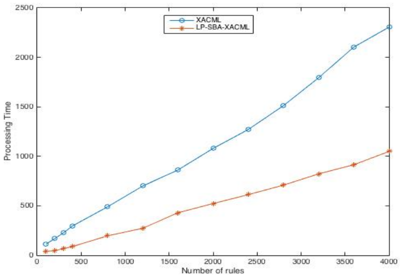


Figure 10: Performance results on mobile

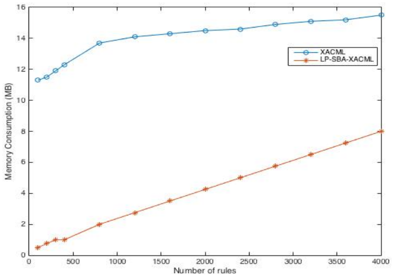


Figure 11: Memory consumption comparison

Table 8: Results of real-world XACML policies on desktop

Number of Rules	XACML	PL-SBA-XACML
2	6	0.2
4	10	0.4
8	14	2.2
16	30	4

Table 9: Results of real-world XACML policies on S3

Number of Rules	XACML	PL-SBA-XACML
2	13	0.9
4	15	1.3
8	20	1.7
16	54	1.9

Table 10: Memory consumption of real-world XACML policies on S3

Number of Rules	XACML	PL-SBA-XACML
2	11.2	0.25
4	11.4	0.25
8	11.7	0.25
16	12.2	0.25

3.6 Conclusion

In this chapter, we addressed the privacy issue that users of IoT devices experience and proposed our privacy constructs that give control back to users. The proposed privacy constructs allow users to set rules and restrictions on when data can be collected. Moreover, we showed performance results of our approach compared to the industry standard XACML in terms of running time and memory consumption. Our approach performed 2-4 times better on constrained device than XACML, whereas in terms of memory consumption, our approach consumed 2-22 times less memory. These results might vary based on policy size and device type

Chapter 4

Deep Learning-based Approach for Activity Recognition and Privacy Customization

4.1 Introduction

In the previous chapter, we've demonstrated the privacy constructs that enable users to control under what conditions their data can be collected. In this chapter, we expand on our previous work to include an intelligent behavior detection model, in which the decision to share private data is determined based on the user's current behavior. In Section 4.2, we elaborate on the process of building our model. Section 4.4 we demonstrate an illustrative scenario and case study. Finally, performance results are shown in Section 4.5.

4.2 Behavior Model Overview

In this section, we introduce our intelligent behavior evaluation module. We explain the steps taken in order to build the model as well as how it integrates with LP-SBA-XACML.

Figure 12 shows the architecture for implementing a personalized machine

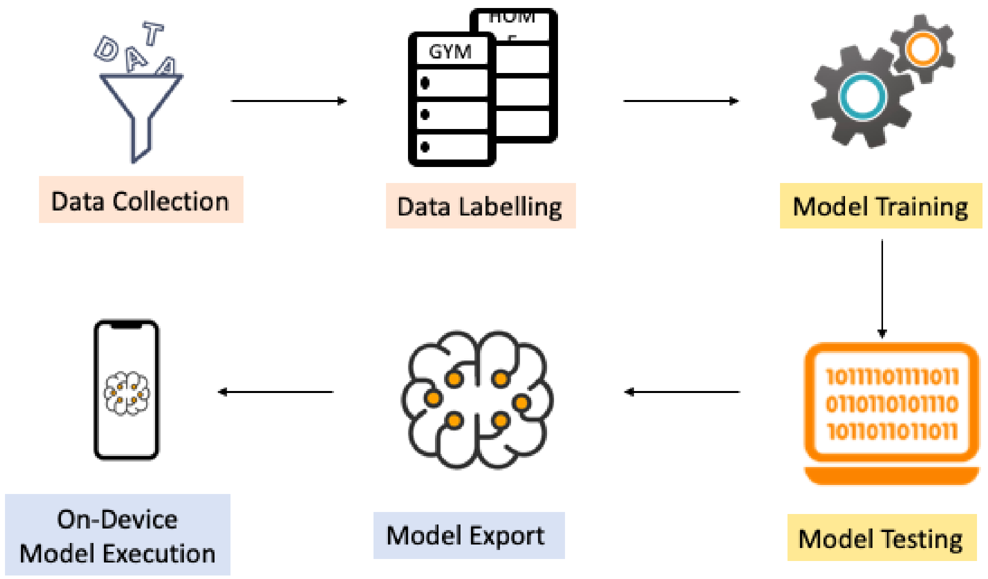


Figure 12: Machine Learning Architecture

learning model.

- Data collection: An android application was created in order to collect the necessary data. The most important features of the collected data are the activity e.g. walking sitting etc. and context such as time and location. The developed application runs in the background without any user interference and collects a record of data every 5 seconds.

- Data Labelling: Upon extracting the text file, we had perform some pre-processing and data labeling. The preprocessing phase was fairly simple, we removed missing records (these were relatively few and do not affect the distribution of the data) and extracted additional useful features such as determining if the current day is a weekend or not (users behave differently on their day off). Finally, since our approach is based on supervised machine learning, the dataset had to be manually labelled. In order to manually infer the behavior, the activity of the user combined with context data result in a high level overview about the current behavior. Moreover, analysis of the activity changes over a period of time had to be performed in order to accurately label the data. For example, if the user was sitting, then walked, then was sitting again, the behavior should be considered the same. This sequencing is important since a record by itself is not enough in order to infer behavior. A summary of the labels is defined in table 1

- Model training: Upon having clean and labeled data, we are ready to create our model. As mentioned earlier, Google's TensorFlow framework was used in order to implement a deep neural network. In order to train the model, the standard train/test split cross validation. Which is a technique that splits the data into training and testing sets. A common practice is to divide the dataset into 80/20, where 80% of the data is used for training and 20% for testing. Finally, during training, cross-validation was used. That is, during each iteration, the model was tested in order to get an overview of how it's performing as it progresses.

- Model testing: Since the data was split into train and test sets, the remaining 20% of data is used in order to evaluate the model's performance. There are several ways to evaluate how a model performs, and one of the most common used metrics is accuracy. For that, our model achieved 90% accuracy on the new unseen test data.

- Model export: After training and testing the model, it's time to export it in order to test it in real-world environment.

- On-Device Model Execution: Once the model was exported, we had to build a sample application that will run the TensorFlow model, collect data in real time, feed the data to he model, and display the output. We've tested the model in the real world in order to get a general idea of how it's performing. The model was successful in accurately inferring our current behavior in real-time. When driving, the model was able to differentiate whether the behavior was commute_to_work, commute_home or outing, thanks to the activity (driving) and to the collected context data (time, location, type of day).

The model was implemented using the TensorFlow framework. [3], which is an open source deep neural network framework implemented by Google. TensorFlow models can run on wide array of devices from servers all the way down to mobile and IoT devices.

Table 1: Labels and record count

Label	Number of records
commute_to_work	1182
grocery_shopping	385
gym	4125
home	27165
outing	5850
sleeping	15567
walk_break	5582
walk_gym	1079
walk_home	1099
work	23364

4.3 Formal Semantics for Behavior Construct

In this section, we present the formal semantics for our behavior construct.

Rules 7 and 8 represent the matching semantics for BC. Simply put, if the user's current behavior (which is inferred using our machine learning model), $\in$ to the set of defined behaviors (BC), then BC matches to true. If, on the other hand, the inferred behavior does $\notin$ to the set, then BC matches to false.

The method **PRED_BEH()** is a function that takes as parameters location, time, network, user activity, and additional features. The method communicates with the behavior module, passing all the necessary parameters in order to infer the user's current behavior. Finally, once the behavior is inferred, the method returns the value and the necessary comparisons are made in order to properly reach a decision.

The formal semantics for **PRED_BEH()** are defined as follows:

$$\{PRED\_BEH, \{Location, Time, Network, UserActivity\}\}$$

Where **PRED_BEH()** is the name of the method to be called, followed by the list of parameters. The model and algorithm are presented in Section 4.2

Table 2: Matching Semantics for Behavior

$$\frac{\exists beh \in BC; PRED\_BEH() = beh}{< B, BC > \underset{match}{\vdash} True} \quad \textbf{(Rule 7)}$$

$$\frac{\forall beh \in BC; PRED\_BEH() \neq beh}{< B, BC > \underset{match}{\vdash} False} \quad \textbf{(Rule 8)}$$

4.3.1 Integrating Behavior Semantics in Privacy Construct

Finally, we integrate the newly proposed behavior semantics into our privacy construct as follows:

Table 3: Evaluation Semantics for PRIV_CONST

$$\frac{(L \underset{match}{\vdash} True) \wedge (N \underset{match}{\vdash} True) \wedge (T \underset{match}{\vdash} True) \wedge (BC \underset{match}{\vdash} True)}{< PRIV\_CONS, R > \underset{eval}{\longrightarrow} PermitObl} \quad \textbf{(Rule 9)}$$

$$\frac{(L \underset{match}{\vdash} False) \vee (N \underset{match}{\vdash} False) \vee (T \underset{match}{\vdash} False) \vee (BC \underset{match}{\vdash} False)}{< PRIV\_CONS, R > \underset{eval}{\longrightarrow} Deny} \quad \textbf{(Rule 10)}$$

Where in order for PC to evaluate to **PermitObl**, all of the premises, T, L, N and BC (behavior condition) must evaluate to true. If, on the other hand, any of the premises matches to false, then PC would evaluate to **Deny** We note that the evaluation semantics for the policy and rule are not affected by this alteration.

4.4 Illustrative Scenario and Case Study

In this section, we present two practical applications for our work:

- Automated Privacy Management: Usage of private data and resources are dynamically managed depending on the user's current behavior.
- Service Management: Services can subscribe to a particular behavior such that, upon the defined behavior is detected, the service is automatically launched.

4.4.1 Automated Privacy Management

Consider the following scenario:

- Operations on sensitive data are prohibited whenever the user is at work
- Operations on sensitive data during weekend are permitted.
- Location cannot be shared if the behavior is one of the following: commute_to_work, work, walk_break. Such a policy is of high importance for confidential locations such as a military base.
- Allow location tracking during weekends
- Block camera at work premise
- Access camera when open and location type is museum
- Audio control is permitted if the detected behavior is commute_to_work
- Phone calls, messages and audio services are blocked when the user is sleeping.

Listing 4.1 represents the corresponding LP-SBA-XACML code to enforce the mentioned requirements.

Listing 4.1: LP-SBA-XACML for Accessing IoT data

```
[1]. PS::=<PS,{PData,PLocation,PGeneral},{PData>PLocation>PGeneral},{deny-
    overrides},{},{},{{},{},{}}>
[2]. P::=<PData,{RWork,RWeekend },{RWork>RWeekend},{deny-overrides
    },{{},{},{}}>
[3]. R::=<RWeekend,{{infer-behavior, {is-weekend}}},{{},{},{}},{permit}>
[4]. R::=<RData,{{infer-behavior, {is-work}}},{{},{},{}},{deny}>
[5]. P::=<PLocation,{LShare,LTrack},{LShare>LTrack},{deny-overrides
    },{},{},{{},{},{}}>
[6]. L::=<LShare,{{infer-behavior, {work}}},{{},{},{}},{deny}>
[7]. L::=<LTrack,{infer-behavior},{is-weekend},{permit},{{},{},{}}>
[8]. P::=<PCamera,{RBlock,RAR},{RBlock>RAR},{deny-overrides},{{},{},{}}>
[9]. R::=<RBlock,{{infer-behavior, {work}}},{deny},{{},{},{}}>
[10]. R::=<RAR,{infer-behavior,{museum}},{{},{},{}},{permit}>
[11]. P::=<PAudio,{RSleep,RAudio},{RSleep>RAudio},{deny-overrides},{{},{},{
}}>
[12]. R::=<RSleep,{infer-behavior,{is-sleeping}},{{},{},{}},{permit}>
[13]. R::=<RAudio,{infer-behavior,{commute-to-work}},{{},{},{}},{permit}>
```

Line 1 corresponds to the policy set, PS, containing three policies: PData, PLocation and PAudio. PData has the highest precedence order followed by PLocation and finally PAudio. There are no obligations for PS and an empty target. Line 2 is the first policy, PData, which is composed of two rules: RWork and RWeekend, where RWork has a higher precedence order than RWeekend. PData has a rule-combining algorithm set to deny-overrides and an empty target. Line 3 is the first rule, RWeekend, which enforces user's requirement of sharing possible sensitive resources, regardless of the behavior, as long as it's during the weekend. The following rule condition has been set:

$$\{infer - behavior, \{is - weeeknd\}\}$$

Where the method infer-behavior gets called, sends a request to the behavior application, where the model executes and infers user's current behavior in real time. Note, however, that if the user happens to be at work, that is, the inferred behavior is **work**, even during weekend, the rule will not be enforced. Finally, we have the rule effect set to permit. Line 4 is the rule that restricts access to sensitive information whenever the user's behavior is: work, commute_to_work, walk_break. The following rule condition is set in order to achieve the desired result:

$$\{infer - behavior, \{is - work\}\}$$

Where infer-behavior is the same method used before, however, we pass an additional parameter is-work, so that the method would check for that particular type of behavior. In case the inferred behavior matches with the condition, then the rule is enforced and a **deny** effect is returned.

Line 5 is a new policy for location, having PLocation as ID. The policy is composed of two rules: LShare and LTrack with LShare having a higher precedence order. The rule combining algorithm has been set to deny-overrides and an empty target.

Line 6 is the location rule that enforces the condition of location sharing, having the location condition set to:

$$\{infer - behavior, \{is - work\}\}$$

Where the method infer-behavior is the same as before, however, the additional parameter, is-work, is passed in order to detect all work-related behaviors. In case the condition is met, then a deny effect is returned. Line 7 is another location rule, LTrack, that enabled tracking of user location based on the following condition:

$$\{infer - behavior, \{is - weekend\}\}$$

Where location tracking is enabled during weekend and a permit effect is returned. Note, however, similar to before, if the inferred behavior is work even during weekend, then the condition is no longer valid.

Line 8 is the policy for camera, PCamera, composed of two rules: RBlock and RAR, where RBlock has a higher precednce order. A rule combining algorithm of deny-overrides is set and an empty target. Line 9 is the RBlock rule, which restricts camera usage whenever the user is within the work premise. The following rule condition has been set:

$$\{infer - behavior, \{camera - work\}\}$$

Where infer-behavior is the same method mentioned earlier. In this case, the parameter camera-work is passed. In case any work-related behavior has been inferred, then the condition is met, and a deny effect takes place.

Line 10 is the RAR rule, which is rule for augmented reality (AR). Camera usage is granted if the user opens the camera and the location that s/he is at is a museum. In this rule, a service provides additional information and guidance within the museum. The following condition has been set to enforce such a rule:

$$\{infer - behavior, \{museum\}\}$$

Where the parameter museum is passed as context information. If the condition has been met, then a permit effect is returned and the AR service is provided to the user. Note that whenever the user exits the museum, even if the camera is still open, the provided service is killed immediately.

Line 11 is the final policy, PAudio, for managing audio related functionalities. PAudio is composed of two rules: RSleep and RAudio, with RSleep having a higher precedence order. Finally, the rule combining algorithm is set to deny-overrides and an empty target. Line 12 is the rule RSleep that enforces the "do not disturb" functionality when the user is sleeping. The following condition has been set to:

$$\{infer - behavior, \{is - sleeping\}\}$$

Where the parameter is-sleeping is passed. In case the condition is met, a permit effect is returned and the service provides the necessary functionalities. Note that once the behavior changes, the service no longer functions.

Finally, line 13 is the last rule, RAudio, that controls audio functionality on the device whenever the user is commuting to work. The following condition has been set:

$$\{infer - behavior, \{commute - to - work\}\}$$

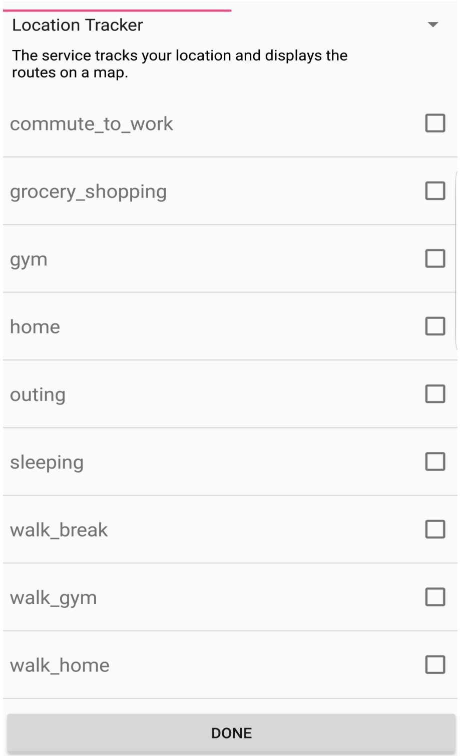
Where the parameter commute-to-work is passed. If the condition is met, then a permit effect is returned. Otherwise, the service will not function.

4.4.2 Service Management

The user is provided with a list of services, and the user selects which service "binds" to a particular behavior. That is, the service will be called and activated once the selected behavior is detected. In this case, service providers work normally, however, the user can customize the usage of the service. Additionally, the service cannot detect when it is launched, that is, which behavior is this service selected for. That way, the privacy of the user is maintained. Consider the following services that wish to register on the user's device:

- S1: Song management that is activated whenever user behavior is "gym" or "commute_to_work"
- S2: Location tracking that is activated during weekend, regardless of the behavior
- S3: Phone call behavior tracking, activated whenever a call is made or received, as long as user is not at work.
- S4: Do not disturb mode when user is busy at work (10-12, 2-3). This service would block all phone calls and messages depending on the time the user is "busy" and automatically stops soon after.

A list of services is displayed to the user, upon selecting one, a detailed page consisting of sub services (if any) is displayed so that the user can customize its usage based on his/her behavior. As mentioned earlier, the service provider is not aware of how the service is used, that is, the mapping between the service and the behavior is known only to the user. That way, privacy is maintained without any information leakage. Figure 13 shows a sample application, where the user matches a service with one or more behavior. Thus, once the behavior is detected, the service is automatically launched.



Location Tracker

The service tracks your location and displays the routes on a map.

- commute_to_work □
- grocery_shopping □
- gym □
- home □
- outing □
- sleeping □
- walk_break □
- walk_gym □
- walk_home □

Figure 13: Behavior Based Service Management

4.5 Experiments and Analysis

In this section, we demonstrate performance tests in order to evaluate how well our model performs. The section is divided into two broad categories: (1) An overview of our TensorFlow model and (2) Performance tests where we show the time taken to infer user's behavior in real time against a request.

4.5.1 TensorFlow Model Overview

This section explores:

- Dataset

- TensorFlow model
- Model performance on mobile

Dataset Overview

Table 1 displays the defined behaviors and the record count for each. In general, the more data we have for each record, the better the model will be at accurately inferring the correct behavior. However, since data collection was performed in, mostly, an uncontrolled environment, it is nearly impossible to evenly collect records for different behaviors.

Model Overview

The structure of our neural network is as follows:

- Input layer: In this layer, the number of neurons must match the number of features. Therefore, the number of neurons was set to 9
- Hidden layer/s: There isn't a specific rule as to how many hidden layers to have, as well as number of nodes. After several trials, we found that one hidden layer with nine neurons gave the best result. Finally, the Rectified Linear Unit (ReLU) was used as an activation function.
- Output layer: The number of neurons in this layer is equivalent to the number of labels, which is 10, using Softmax activation function.

Additional parameters need to be set in a neural network:

- Optimizer: Adam
- Metrics: Accuracy
- Loss function: Sparse categorical cross entropy
- Learning rate: 0.0001

- Epochs: Number of iterations to be made, our model was found to converge at 1000 epochs.
- Batch size: This parameter depends on how powerful your machine is. Usually, the higher the batch size the faster it is to train the model. On our machine, batch size of 2048 was found to be optimal.

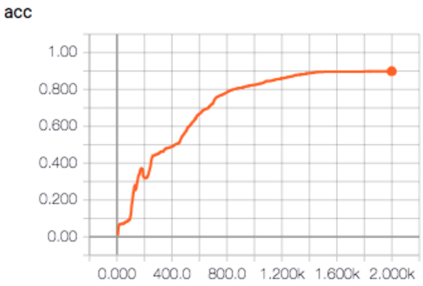


Figure 14: Accuracy

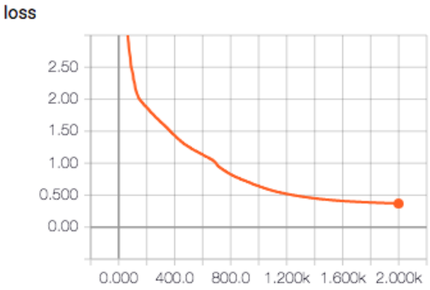


Figure 15: Loss

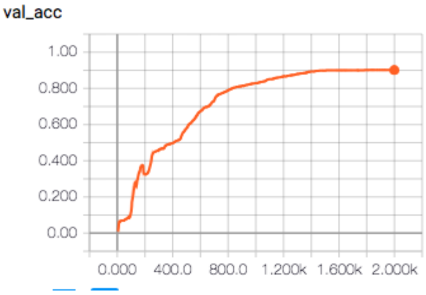


Figure 16: Validation Accuracy

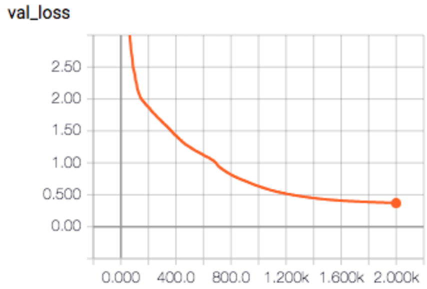


Figure 17: Validation Loss

The data was split 80-20, 80 for training and 20 for testing. When training the model, upon the completion of each epoch, cross validation was made in order to get an overview how the model is performing.

Upon checking figure 14, it can be seen that the model's accuracy continuously increases, with few "bumps" at 200-300 epochs. Afterwards it steadily improves until hitting a plateua, reaching 90% accuracy. Increasing the number of epochs at this stage might yield to better results in terms of accuracy, however, you risk overfitting the model. You want the model to capture the "trend" and pattern of the data in order to generalize well.

Similarly, figure 16, which shows the curve for validation accuracy, shows similar results. This indicates that the model manages to generalize in each epoch.

Figure 15, which displays the model loss. Recall that the used loss function is sparse categorical cross entropy. From the figure, it shows the gradual decrease of loss with each iteration. Notice that there is no "steep" decline, indicating that the model is incrementally improving. Finally, as with accuracy, the loss reaches a plateau at the end of iterations. The same loss results occur for validation, shown in figure 17.

4.5.2 Performance Tests

Table 4 summarizes performance runs performed on Samsung Galaxy S8. The tests were performed as follows:

Table 4: TensorFlow Performance Runs on S8

Number of predictions	Time(ms)
100	14.45
200	22.43
300	33.22
400	40.34
800	69.58
1200	103.1
1600	132.38
2000	164.62
2400	188.53
2800	226.22
3200	260.14
3600	292.28
4000	321.4

We simulated x number of requests, and for each request, we make an inference on a test record. The results show that the model scales well up to 4000 requests, where it requires approximately 320ms. Considering the type of application for such a model, it's highly unreasonable to receive such high number of requests. Furthermore, performance can be drastically improved by implementing one of the following approaches:

- Parallelization: We can create a ThreadPool composed of ten threads, and divide the number of requests evenly among the created threads. Theoretically, this should drastically improve performance, ideally up to ten fold.

- Periodic predictions: A prediction is performed once every 5 seconds. Thus, regardless how many requests are received, the same behavior is used. Thereby drastically reducing the number of inferences performed. Once a request is received after a duration longer than 5 seconds, a new inference is performed.

For the current application, the performance is more than acceptable, especially when considering the unlikeliness of receiving 100 requests (which requires 14.45ms). Creating a TensorFlow model and running it on mobile is not enough, it is of high importance to take into considerations the complexity of the generated model. In our scenario, the generated model is 2KB in size, which is feasible to run on almost any resource constrained device. Additionally, only 9 parameters/features are required to make an inference. The simplicity of the model coupled with the small size makes it ideal for integration in IoT applications.

4.6 Conclusion

In this chapter, we presented our intelligent behavior detection model and how it seamlessly integrates with our privacy policy language. Moreover, we explained how the model was implemented using the TensorFlow framework and performed synthetic and real life experiments to evaluate the performance of our model. Our model achieved an accuracy of 90% for inferring user's behavior. The accuracy can be improved if we have more participants and more labelled data. Moreover, our model runs in real time and offline on a resource constrained device. Added to that the small size of the model (2KB), and performing upto 4000 predictions in 321.4 ms, makes the model practical to run on IoT devices with reasonable perofrmance and accuracy.

Chapter 5

Conclusion

In this thesis, we built a lightweight privacy policy language on top of the work done by [33] in order to provide users with personalized privacy in the IoT era. Moreover, we demonstrated performance results on desktop and resource constrained device and achieved a reasonable performance in terms of time and memory, making it practical to run on IoT devices. Finally, we implemented a smart behavior detect module using the TensorFlow framework to infer user's behavior in real time, and integrated user's behavior with the privacy policies. That way, the condition and constraint in which data is shared is dependent on the user's context and behavior. Finally, synthetic and real world experiments were conducted to show the feasibility of our results.

Based on our attained results, we can branch in several directions for future work. First obvious route is to collect more data from a diverse set of users, we can also try different machine learning algorithms or creating an ensemble algorithm to achieve a better and more accurate result. Moreover, we can identify users based on their behavior and detect anomalies i.e. user's behavior does not match with his/her history. Another possible direction is creating specialized models for specific domains such as banking to monitor employee behavior with minimal intrusiveness. Healthcare is another possibility in which we have a model that monitors users vital signs and recommend a medical advice when some threshold is exceeded. We can even build several sub low level behavior detection models, and stack on top of them our high level model to infer broader and more accurate behavior.

5.1 List of Publications

The following list of publications has been derived from the current thesis work:

5.1.1 Conference

- Mohamad Chehab and Azzam Mourad. "Towards a Lightweight Policy-Based Privacy Enforcing Approach for IoT". *In the Proceedings in The International Conference on Computational Science and Computational Intelligence*, Las Vegas, Nevada, USA.

5.1.2 Draft

- Mohamad Chehab and Azzam Mourad. "Inteligent Behavior Aware User Controlled Privacy for IoT".

Bibliography

[1] *Machine Learning Basics.*

[2] Wso2 balana https://github.com/wso2/balana.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[4] M. M. Aborokbah, S. Al-Mutairi, A. K. Sangaiah, and O. W. Samuel. Adaptive context aware decision computing paradigm for intensive health care delivery in smart citiesa case analysis. *Sustainable Cities and Society*, 41:919–924, 2018.

[5] C. C. Aggarwal. On randomization, public information and the curse of dimensionality. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 136–145. IEEE, 2007.

[6] C. C. Aggarwal and S. Y. Philip. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining*, pages 11–52. Springer, 2008.

[7] N. Apthorpe, D. Reisman, and N. Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[8] H. F. Atlam, A. Alenezi, R. K. Hussein, and G. B. Wills. Validation of an adaptive risk-based access control model for the internet of things. *Inter-*

national Journal of Computer Network and Information Security, 10(1):26, 2018.

[9] H. F. Atlam, A. Alenezi, R. J. Walters, G. B. Wills, and J. Daniel. Developing an adaptive risk-based access control model for the internet of things. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 655–661, 2017.

[10] G. Bai, L. Yan, L. Gu, Y. Guo, and X. Chen. Context-aware usage control for web of things. *Security and Communication Networks*, 7(12):2696–2712, 2014.

[11] F. Beligianni, M. Alamaniotis, A. Fevgas, P. Tsompanopoulou, P. Bozanis, and L. H. Tsoukalas. An internet of things architecture for preserving privacy of energy consumption. 2016.

[12] G. Chen, X. Ding, K. Huang, X. Ye, and C. Zhang. Changing health behaviors through social and physical context awareness. In *2015 International Conference on Computing, Networking and Communications (ICNC)*, pages 663–667. IEEE, 2015.

[13] Y. Chervonyi, D. Harabor, B. Zhang, and J. Sacks. Zap: Making predictions based on online user behavior. *arXiv preprint arXiv:1807.06046*, 2018.

[14] L. Ciabattoni, G. Foresi, A. Monteriù, D. P. Pagnotta, L. Romeo, L. Spalazzi, and A. De Cesare. Complex activity recognition system based on cascade classifiers and wearable device data. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2. IEEE, 2018.

[15] F. Cmara Pereira and S. S.Borysov. *Chapter 2 - Machine Learning Fundamentals*, chapter 2, pages 9–29. Elsevier, 20019.

[16] M. Dabbagh and A. Rayes. *Internet of Things Security and Privacy*, pages 211–238. Springer International Publishing, Cham, 2019.

[17] F. Deng, J. Lu, S.-Y. Wang, J. Pan, and L.-Y. Zhang. A distributed pdp model based on spectral clustering for improving evaluation performance. *World Wide Web*, pages 1–22, 2018.

[18] F. Deng, S. Wang, L. Zhang, X. Wei, and J. Yu. Establishment of attribute bitmaps for efficient xacml policy evaluation. *Knowledge-Based Systems*, 143:93–101, 2018.

[19] F. Deng and L.-Y. Zhang. Elimination of policy conflict to improve the pdp evaluation performance. *Journal of Network and Computer Applications*, 80:45–57, 2017.

[20] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[21] A. Friedman and A. Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–502. ACM, 2010.

[22] Gartner. Gartner says 8.4 billion connected.

[23] P. Golle, F. McSherry, and I. Mironov. Data collection with self-enforcing privacy. *ACM Transactions on Information and System Security (TISSEC)*, 12(2):9, 2008.

[24] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, 58(5-6):1189–1205, 2013.

[25] N. P. Hoang and D. Pishva. A tor-based anonymous communication approach to secure smart home appliances. In *Advanced Communication Technology (ICACT), 2015 17th International Conference on*, pages 517–525. IEEE, 2015.

[26] X. Huang, R. Fu, B. Chen, T. Zhang, and A. Roscoe. User interactive internet of things privacy preserved access control. In *Internet Technology And Secured Transactions, 2012 International Conference for*, pages 597–602. IEEE, 2012.

[27] E. Kidmose, E. Ebeid, and R. H. Jacobsen. A framework for detecting and translating user behavior from smart meter data. *arXiv preprint arXiv:1807.03111*, 2018.

[28] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle. Dtls based security and two-way authentication for the internet of things. *Ad Hoc Networks*, 11(8):2710–2723, 2013.

[29] J. H. Lee, J. Shin, and M. J. Realff. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, 114:111–121, 2018.

[30] A. X. Liu, F. Chen, J. Hwang, and T. Xie. Xengine: a fast and scalable xacml policy evaluation engine. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 265–276. ACM, 2008.

[31] J. Liu, Y. Xiao, and C. P. Chen. Authentication and access control in the internet of things. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 588–592. IEEE, 2012.

[32] T. Mashita, D. Komaki, M. Iwata, K. Shimatani, H. Miyamoto, T. Hara, K. Kiyokawa, H. Takemura, and S. Nishio. A content search system for mobile devices based on user context recognition. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 1–4. IEEE, 2012.

[33] A. Mourad and H. Jebbaoui. Sba-xacml: Set-based approach providing efficient policy decision process for accessing web services. *Expert Systems with Applications*, 42(1):165–178, 2015.

[34] A. Narayanan, E. Shi, and B. I. Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1825–1834. IEEE, 2011.

[35] G. Neagu, V. Florian, A. Stanciu, and S. Preda. Sensing as a service approach in health monitoring. In *RoEduNet Conference: Networking in Education and Research, 2016 15th*, pages 1–5. IEEE, 2016.

[36] V. Oleshchuk. Internet of things and privacy preserving technologies. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, pages 336–340. IEEE, 2009.

[37] H. Omrani. Predicting travel mode of individuals by machine learning. *Transportation Research Procedia*, 10:840–849, 2015.

[38] J. Qi, P. Yang, D. Fan, and Z. Deng. A survey of physical activity monitoring and assessment using internet of things technology. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 2353–2358. IEEE, 2015.

[39] R. I. Ramos-Garcia and A. W. Hoover. A study of temporal action sequencing during consumption of a meal. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, page 68. ACM, 2013.

[40] T. A. Rath and J.-N. Colin. Adaptive risk-aware access control model for internet of things. In *2017 International Workshop on Secure Internet of Things (SIoT)*, pages 40–49. IEEE, 2017.

[41] A. Rayes and S. Salam. *Internet of Things (IoT) Overview*, pages 1–35. Springer International Publishing, Cham, 2019.

[42] I. H. Sarker, M. A. Kabir, A. Colman, and J. Han. An improved naive bayes classifier-based noise detection technique for classifying user phone call behavior. In *Australasian Conference on Data Mining*, pages 72–85. Springer, 2017.

[43] I. H. Sarker and F. D. Salim. Mining user behavioral rules from smartphone data through association analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 450–461. Springer, 2018.

[44] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4):426, 2016.

[45] M. Shoaib, H. Scholten, and P. J. Havinga. Towards physical activity recognition using smartphone sensors. In *Ubiquitous intelligence and computing, 2013 ieee 10th international conference on and 10th international conference on autonomic and trusted computing (uic/atc)*, pages 80–87. IEEE, 2013.

[46] S. Tim. Internet of things (iot) data continues to explode exponentially. who is using that data and how? `https://blogs.cisco.com/datacenter/internet-of-things-iot-data-continues-to-explode-exponentially-who-is-usin` 2018.

[47] J. P. Varkey, D. Pompili, and T. A. Walls. Human motion recognition using a wireless sensor-based wearable system. *Personal and Ubiquitous Computing*, 16(7):897–910, 2012.

[48] P. Vepakomma, D. De, S. K. Das, and S. Bhansali. A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In *BSN*, pages 1–6, 2015.

[49] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 2018.

[50] B. D. Weinberg, G. R. Milne, Y. G. Andonova, and F. M. Hajjat. Internet of things: Convenience vs. privacy and secrecy. *Business Horizons*, 58(6):615–624, 2015.

[51] G. M. Weiss, J. W. Lockhart, T. T. Pulickal, P. T. McHugh, I. H. Ronan, and J. L. Timko. Actitracker: a smartphone-based activity recognition system for improving health and well-being. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 682–688. IEEE, 2016.

[52] E. Wu, P. Zhang, T. Lu, H. Gu, and N. Gu. Behavior prediction using an improved hidden markov model to support people with disabilities in smart homes. In *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*, pages 560–565. IEEE, 2016.

[53] J. Zhao, T. Jung, Y. Wang, and X. Li. Achieving differential privacy of data disclosure in the smart grid. In *INFOCOM, 2014 Proceedings IEEE*, pages 504–512. IEEE, 2014.

[54] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *ICNP*, volume 14, pages 221–232, 2014.

[55] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, and J. Gao. Atrank: An attention-based user behavior modeling framework for recommendation. *arXiv preprint arXiv:1711.06632*, 2017.