UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA - VARESE

DiSTA

Dipartimento di Scienze Teoriche e Applicate

PHD THESIS

to obtain the title of

Doctor of Science

Specialty : Computer Science

Defended by

Bikash Chandra Singh

PRIVACY PREFERENCE MECHANISMS IN PERSONAL DATA STORAGE (PDS)

Advisor: Prof. Barbara Carminati

Advisor: Prof. Elena Ferrari

defended on October 05, 2018

Jury :

Reviewers :	Prof. Athena Vakali, Aristotle University of Thessaloniki, Greece
	Prof. Georgia M. Kapitsaki, University of Cyprus, Cyprus
President :	Prof. Elena Ferrari, Università degli Studi Dell'Insubria, Italy
Examinators :	Prof. Pierluigi Gallo, Università degli Studi di Palermo, Italy
	Prof. Claudio Agostino Ardagna, Universita' degli Studi di Milano, Italy

Dedication.....
To my mother Nioti Rani Singh
To my father Haresh Chandra Singh
To my wife Supreya Singh Roy
To my sister Lucky Rani Singh
To my brother Ripon Kumar Singh

Acknowledgments

I would like to utilize this opportunity for expressing my gratitude to all who supported me to achieve this goal.

First of all, I would like to thank the Almighty God for blessing me to accomplish this thesis. Without His blessing, I could not able to continue this work.

I would like to express my deepest gratitude to my advisors Prof. Elena Ferrari and Prof. Barbara Carminati for their continuous support, guidance, effort, and engagement throughout this research over the years which gave me enormous strength to pursue this thesis successfully. Specially, their patience, motivation, immense knowledge, and enthusiasm greatly enriched my research work. This thesis would not have been possible without the support of my advisors. I feel honor being one of their PhD students.

I would like to acknowledge my thesis committee members: Prof. Athena Vakali, Department of Computer Science, Aristotle University of Thessaloniki and Prof. Georgia M. Kapitsaki, Department of Computer Science, the University of Cyprus for their enlightening comments and encouragement that inspire me to thought on different aspects of the work presented in this thesis.

Next thanks go to my lab partners Gökhan Sağırlar, Pietro Colombo, Davide Albertini, Tú Hoàng Anh, Boffa Stefania, Md. Zulfikar Alom, Vergani Alberto Arturo, Naeimeh Laleh, Leila Bahri, Ngoc Hong Tran, and Cüneyt Gurcan Akçora. We have passed some great moments together. They made my PhD days so enjoyable and memorable.

I would like to thank for all the people who take participations in my experiments. I really appreciate their efforts and time for helping me to evaluate my research work.

Last but not least, I would like to thank my family members. Without their endless supports and love I could not finish this thesis successfully. I am very grateful to them.

Abstract

In this thesis, we study frameworks for managing user's privacy when disclosing personal data with third parties from Personal Data Storage (PDS). PDS is a secure digital space which allows individuals to collect, store, and give access to third parties. So, PDS has inaugurated a substantial change to the way people can store and control their personal data, by moving from a service-centric to a user-centric model. Up to now, most of the research on PDS has focused on how to enforce user privacy preferences and how to secure data stored into the PDS. In contrast, this thesis aims at designing a *Privacy-aware Personal Data Storage* (P-PDS), that is, a PDS able to automatically take privacy-aware decisions on third parties access requests in accordance with user preferences. This thesis first demonstrates that semi-supervised learning can be successfully exploited to make a PDS able to automatically decide whether an access request has to be authorized or not. Furthermore, we have revised our first contribution by defining strategies able to obtain good accuracy without requiring too much effort from the user in the training phase. At this aim, we exploit active learning with semi-supervised approach so as to improve the quality of the labeled training dataset. This ables to improve the performance of learning models to predict user privacy preferences correctly.

Moreover, in the second part of the thesis we study how user's contextual information play a vital role in term of taking decision of whether to share personal data with third parties. As such, consider that a service provider may provide a request for entertainment service to PDS owner during his/her office hours. In such case, PDS owner may deny this service as he/she is in office. That implies individual would like to accept/deny access requests by considering his/her contextual information. Prior studies on PDS have not considered user's contextual information so far. Moreover, prior research has shown that user privacy preferences may vary based on his/her contextual information. To address this issue, this thesis also focuses to implement a contextual privacy-aware framework for PDS (CP-PDS) which exploits contextual information to build a learning classifier that can predict user privacy preferences under various contextual scenarios. We run several experiments on a realistic dataset and exploiting groups of evaluators. The obtained results show the effectiveness of the proposed approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays personal data are excerpted by many information systems, such as health-care, banking, e-commerce services etc., for providing online services [17]. Individuals are gradually becoming dependent on these online services given the benefits in easing their daily life activities (e.g., shopping, treatment). Despite their benefits, online services require lot of personal data, even more than those strictly required for providing services. By providing personal data to third parties, individuals are loosing control on them. Moreover, online services exchange data among them, making even more difficult for an individual to keep the control on his/her data. Furthermore, nowadays personal data are stored into the repository of service providers but this has many drawbacks on data privacy. One of the most relevant is that such a paradigm on the one hand relies on the assumption that the service provider is fully trusted and, on the other hand, prevents users from fully exploiting their data, since each provider keeps a separate view of them. More precisely, we have to completely trust the provider with respect to the release and sharing of our personal data. Therefore, it is difficult for a user to trace which of his/her data are used/shared by providers and for which purposes and also to fully understand the privacy implication of personal data release. A clear example of this risk is given by the recent case of Cambridge analytica, where Facebook has shared a large amount of personal data with this company for not well-declared purpose [32, 92].

Figure 1.1 shows how currently service providers collect individuals personal data in order to provide them online services. For instance, user A has to provide his/her personal data to several service providers (e.g., hospital, bank, social media, etc.) having then his/her data stored and replicated into each single provider's repository. In this way, user A loses the control on his/her personal data as well as service providers can violate the user trust, as an example, by sharing his/her data with other third parities. So, at present, this service-centric data storage paradigm arises a big problem for ensuring user privacy on their personal data.

In an attempt to overcome this situation, researchers are working on the creation of unified repositories of personal data so that individuals can handle their data properly as well as share the data anywhere they want [9, 46, 71, 101]. As a matter of fact, individuals

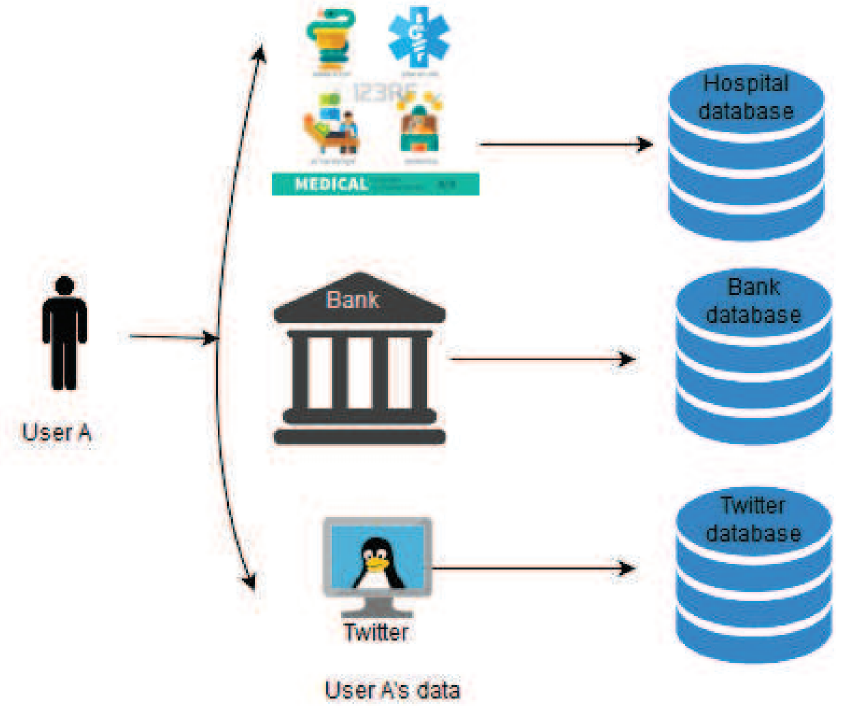


Figure 1.1: Data stored in service providers' databases

interact with various online service systems according to a service-centric data storage paradigm, where data of individuals are stored by and under control of services [91]. The main idea behind Personal Data Storage [6, 35, 91] is to store personal data to be shared with different data sources into a unique logical repository under the control of end users. This view is also enabled by recent developments in privacy legislation and, in particular, by the new EU General Data Protection Regulation (GDPR), whose art. 20 states the right to data portability, according to which "the data subject shall have the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format", thus making possible data collection into a PDS. Mainly, GDPR gives control to individuals over their personal data whether they want to share or not with third parties. Moreover, it simplifies the regulatory environment for business purpose by unifying the regulation within the EU [4]. In addition, the Cloud Security Alliance (CSA) provides Code of Conduct for GDPR Compliance offers a consistent and comprehensive framework that can help service providers for complying with the European Union's GDPR. Moreover, the CSA Code of Conduct offers a compliance tool - Privacy Level Agreement (PLA) that provides guidelines regarding the level of data protection offered by a cloud service provider with a mechanism so as to identify a baseline of mandatory personal data protection legal requirements across the European Union (EU).

Figure 1.2 shows the concept of personal data storage (PDS). According to PDS ar-

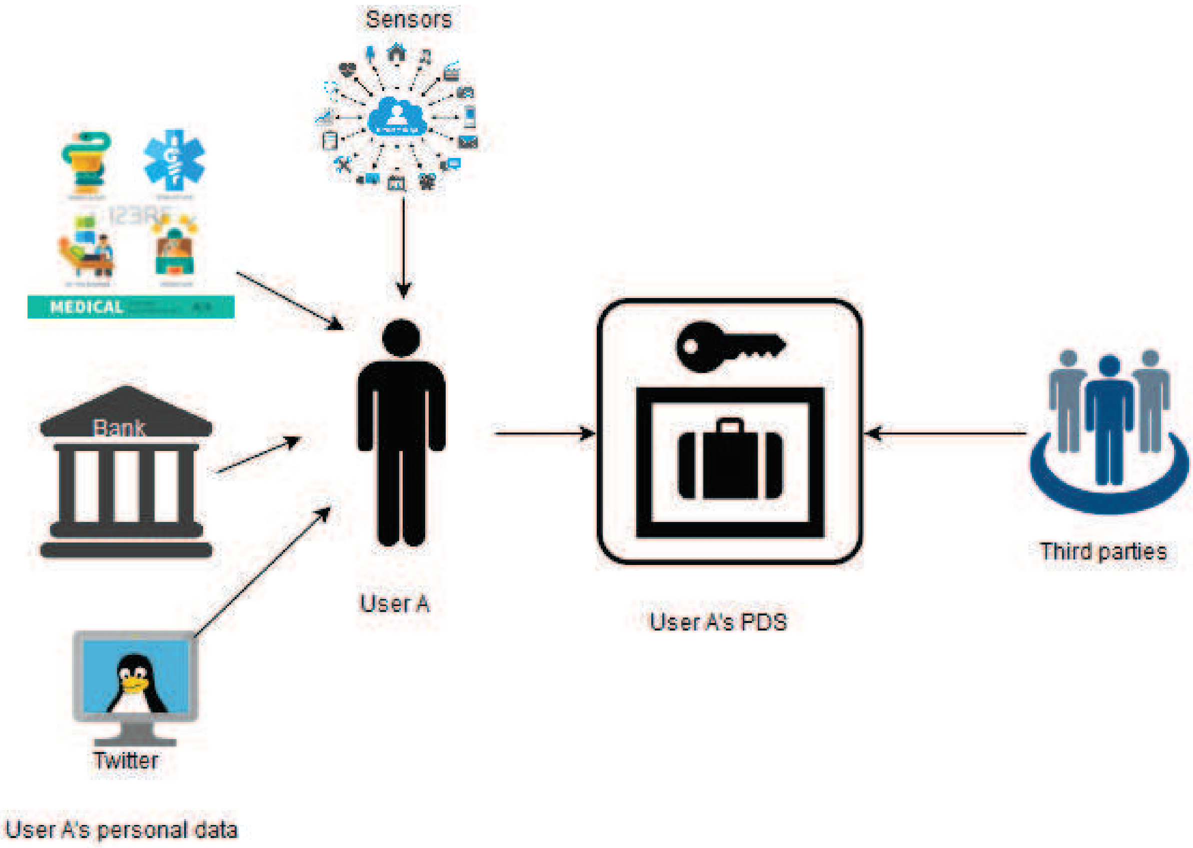


Figure 1.2: The concept of Personal Data Storage (PDS)

chitecture, for instance, user A's personal data is stored into his/her PDS controlled, in terms of sharing data with third parties, directly by user A. In addition, we can consider cloud storage for PDS so as individual's personal data can be stored in a remote location that can be accessed by the permission of the owner. This model can be considered as a user-centric storage model [91], where data owners can control their own data. The data stored into PDS can be used for further usages. In this way, users have control to share their personal data with third parties.

Although the PDS paradigm ensures that individuals have strong control on personal data, an important issue is *how users can easily set their privacy preferences* on personal data release according to their privacy requirements. Interestingly, several studies have shown that *average users might have difficulties in properly setting potentially complex privacy preferences* [3, 59, 63, 89]. For instance, Facebook offer a privacy setting page that allows users to specify their own privacy preferences but most of the users can not properly set up privacy preferences according to their requirements [61]. Thus, this is a challenging job to design a privacy preference framework for end users since most of the cases users are not so expert to configure their privacy preferences properly. Furthermore, users are getting tiredness or feeling bore if they are going to set up their privacy preferences manually [61]. However, there is no proper tools that can help users to set up their privacy preferences in PDS yet.

Thus, the main goal of this dissertation is to study and design privacy preference mechanisms that help users to set their privacy preferences in PDS. To do so, we have exploited machine learning tools so that our proposed privacy preference mechanisms can automatically set up privacy preferences based on user activities. We also proposed new approaches with active learning to improve the prediction of user privacy preferences correctly on access requests.

1.1 Contributions

The main contributions of this thesis lies in (1) designing, and modeling the architecture for privacy-aware personal data storage, and (2) studying, designing, prototyping, and evaluating the performance of mechanisms to automatically set up PDS user privacy preferences. In summary, this thesis provides the following research contributions:

1. We propose a new technical approach that empowers individuals to have better control on their personal data in PDS. Particularly, we present a privacy-aware PDS architecture by focusing on two logical data zones based on the categories of personal data. Moreover, we present a privacy-aware PDS that set privacy preference strategies based on risks and benefits in term of data release with respect to the data owners perspective.

2. We propose different learning algorithms that allow a fine-grained learning of the privacy aptitudes of PDS owners. More specifically, it has been demonstrated that semi-supervised learning can be successfully exploited to make a PDS able to automatically decide whether an access request has to be authorized or not. To the best of our knowledge, we are the first to exploit the correlations among the dimensions of an access request elements that impact individual decisions so as to build up the learning models. The learned models are then used to predict the class label of third party access requests by exploiting user feedbacks on a training dataset.

3. This thesis also enhances the focus on the optimization for the learning processes so as to have a more usable PDS, in terms of reducing the effort for the training phase, as well as a more conservative approach w.r.t. users' privacy, when learning models produce conflicting decisions on access requests. At this aim, our studies deeply revise the learning process of active learning (AL) approach that tunes the uncertainty sampling strategy so as to select more relevant labeled training dataset to improve the performance the learning models.

4. We further extended the learning processes as to not only considering the elements of access requests to train up learning models but also exploiting user contextual data in term of defining privacy preference in PDS. More particularly, the proposed approach

uses the current situation of PDS owner as learning features so as to build privacy learning models. The aim of this extension is to design a contextual based privacy preference model that in presence of information about access request context is able to react so as to have a decision that takes into account also users' preferences w.r.t. the context.

1.2 Outline of the Dissertation

The dissertation is structured into six chapters organized as follows:

Chapter 1 mainly discusses the motivation and the main contributions of this dissertation.

Chapter 2 discusses previous work on privacy preferences on users' personal data. It mainly introduces the preliminary works on Personal Data Storage (PDS) to help users for setting privacy preferences. Afterwards, it summarizes the existing work that exploit user's contextual information so as to build privacy preference models to protect personal data from unauthorized third parties. The chapter closes with the related work on personal data protection in social media domains.

Chapter 3 describes our proposed PDS architecture based on risk-benefit metrics for ensuring user's privacy. The proposed privacy preference mechanism measures the benefit and risk of user in releasing personal data with third parties and compare these values to take final decision.

Chapter 4 describes the various approaches to learn PDS owners' privacy habits. At this purpose, in this chapter, we present three machine learning approaches to predict users opinions on releasing personal data from PDS and the experiments we have done to determine which learning approach provides better accuracy in PDS scenarios.

Chapter 5 illustrates the privacy preference learning approaches that focus to improve the quality of users feedbacks in training phase so as to improve the performance of the proposed approaches. At this purpose, in this chapter, we mainly present an extended version of active learning called as history-based active learning (HBAL) and derive personalized history-based active learning (PHBAL) from HBAL to predict class labels when classifiers produce conflicting decisions on access requests.

Chapter 6 focuses on user's contextual information and defines a learning approach exploiting contextual features to learn user privacy preferences.

Appendix A enlists a list of publications where the research activities described in this thesis are presented.

Chapter 2

Related Work

2.1 Introduction

Our research has focused on proposing privacy preference mechanisms on PDS, thus we will start to discuss the related work that have focused to implement privacy preference mechanisms on PDS using various strategies. Therefore first we discuss the related work that exploit different strategies for privacy preference mechanisms on PDS in Section 2.2. In Section 2.4 we explain machine learning approaches based on non-contextual data that have been used to implement privacy preferences for protecting personal data. After that, we cover those approaches that exploit users contextual data and non-contextual data for learning users privacy preferences to protect personal data from unwanted third parities access in PDS in Section 2.5.

2.2 Privacy preferences on PDS

In order to address the growing problem of spreading personal data over the Internet, researchers have started to propose different approaches that can mitigate this problem. In particular, several approaches have been defined based on PDS to fend the spreading personal data so as individuals able to control their data in term of sharing with others. De Montjoye et al [35] presented openPDS/SafeAnswers mechanism that allows individuals to collect, and store their personal data in PDS as well as give access to their metadata to third parties based on privacy policies. The framework defines a mechanism for returning to third parties only aggregated answers, based on their questions, instead of raw data. Although this framework never shares raw data, there is room for malicious applications to infer more information through a specific sequence of questions-answers, which can eventually breach user privacy. Allard et al [6] proposed a framework to design a personal data server approach and they identify the main technical barriers for implementing PDS and try to find the preliminary solutions. They suggested portable and secure devices with NAND flash chip for storing personal data so that user can fully control how his/her data are shared with others.

Recently, researchers have proposed models for user-centric storage in the cloud domain, where data are stored and controlled by users. For instance, Oort [26] is a user-centric cloud storage system that organizes data by users rather than applications, considering global queries which find and combine relevant data fields from relevant users. Moreover, it allows users to choose which applications can access their own data, and which types of data to be shared with which users. In [74], authors suggest mechanism for secure data storage that are able to store personal data and could be accessed from anywhere using queries using the secret sharing technique. Recently, [114] proposed a Block chain-based Personal Data Store (BC-PDS) framework which leverage on BlockChain to secure the storage of personal data.

Sieve [98] allows user to upload encrypted data to a single cloud storage. It utilizes key-homomorphic scheme to provide cryptographically enforced access control. Amber [27] has proposed an architecture where users can choose applications to manipulate their data-but it does not mention either how the global queries work or how the application providers interact with. In [35], authors developed a user-centric framework that share with third parity only the answers to a query instead of the raw data. Mortier et al. [70] have proposed a trusted platform called Databox, which can manage personal data by a fine grained access control mechanism but do not focus on policy learning. Our privacy-aware PDS is different from other approaches in the matter of taking into account the learning mechanisms that can learn user's privacy preferences from user participations automatically so as to define users privacy requirements in PDS.

2.3 Privacy preferences in social media

Besides PDS, we also focus on the related work on social media privacy preferences. Many researchers have focused on the development of privacy preference models for Online Social Networks (OSNs) [33, 39, 41], where the authors proposed frameworks to share their data with predefined list of friends based on set of rules. In [30], authors stated an access control framework that provides flexible and fine-grained controls on how third party applications (TPAs) can access OSN user's data based on user-specified policies in terms of relationship between the user and the application. In [86], authors proposed a framework that prohibit untrusted applications from leaking users' private information, whereas, in [97] authors enforced sandbox in both server and client sides so as to restrict information flow from users to application developers. [93] proposed a framework that captures user's privacy intentions with clustering approach based on explicit user behavior. The main idea is that once the user adds a friend into a cluster then this friend get permission to access to all data in that cluster. In [7, 43], authors proposed a privacy-preserving online advertising systems that aim to protect user profiles from ad brokers. [51] introduced privacy preference based on different types of service. In [47], authors proposed a model for privacy preference in terms of a hierarchy of questions which can be asked to users. [66] introduced the PViz privacy tool which allows the user to understand the visibility of his/her profile according to natural sub-groupings of friends, and at different levels of granularity. Basically, these works enhanced privacy preference based on the feedback from a user. But as we know that average users are not expert enough to set up their privacy requirement in term of providing appropriate answers in a set of questions, thus these works could not provide well defined mechanisms for user privacy preferences. In fact, prior researchers have shown that users' privacy can sometimes be violated with social networks [38, 40]. In addition, other related work proposed different risk models [2, 50, 53, 83, 99] so that users would be able to understand the privacy risk for sharing their personal data with particular friends or others. Moreover, there are some frameworks which avail the concept of incentive mechanism [100, 110] for sharing data with third parties. [54] explores an game theoretic framework by which they measure the effects of different incentives and potential trust for sharing data. The above privacy preference models provide good performance social networking services but still now, they are being criticized for failing to adequately protect their users' privacy [18].

In our proposal, we focus on to reduce user privacy threat in term of sharing data with third parties but in different way. Our approach exploits semi-supervised learning so as to take minimum feedbacks from users about their privacy preferences on access requests with the aim of reducing user's burden. Our analysis of learning privacy demonstrates that user's decisions on access request have been influenced by the correlation among elements of access requests so as the ensemble learning produces less false negative and positive rates compared to other approaches.

2.4 Privacy preferences learning based on non-contextual data

Non-contextual data refers to the data fields which are not directly related to users present situation. In this section, we focus on the work that exploit non-contextual data for setting privacy preferences. In [52], authors have proposed an approach that optimizes the utility-privacy tradeoff in personalized services as web search. In [55] examines different dimensions of privacy practices that impact user's willingness to permit the collection of data for Online behavioral advertising (OBA). In [108], authors have presented a framework based on the concept of personalized anonymity that takes into account customized privacy requirements. Some research are designed to limit access to personal information within an organization, by deploying privacy-aware access control [69] within that organization can make decisions and enforce access control policies, intercepting queries to data repositories and returning sanitized views (if any) on requested data.

Recently, Nakamura et al. [75] proposed a machine-learning approach to set up user personalized privacy settings for third party access. Similarly to our approach, the proposed learning model is based on information on service providers, type of requested personal data, and usage purposes. In particular, the approach presented in [75] delivers a set of 80 questions to each user at the time of registration to a new service. Among the received answers, the approach repeatedly selects a combination of five questions-answers as training data, and use supervised multiclass SVM [44] to learn individual privacy settings.

Then, the combination with the best accuracy is selected. Our approach differentiates from work in [75] along several directions. First, we adopt semi-supervised approaches (i.e., Expectation-Maximization (EM)-based algorithms) so as to reduce the user burden on getting training dataset. As it will be presented in Section 4.4, EM-based approaches provide a better accuracy than SVM with the same training set. Moreover, we obtain accuracy values similar to those obtained in the proposal in [75] but with only 40 initial questions as training dataset, whilst [75] requires 80 for each registered service. Furthermore, we explored different learning algorithms to find the best fitting one but also to investigate how correlations among access request features impact the individual decision process. Proposal in [75] only considered SVM. [8, 34, 39] have investigated the use of semi-supervised and unsupervised approaches to automatically extract privacy settings in social media. In [80], authors have mentioned that location based data can be considered contextual data. They have compared the accuracy of manually set privacy preferences with the one of an automated mechanism based on machine learning. The results show that machine learning approaches provide better result than user-defined policies.

The issue of privacy preference suggestion has also been investigated in other contexts, such as for instance social networks. Indeed, almost all social media have a privacy setting page to allow users to set up their privacy preferences. However, studies have shown that users are facing many problems in the privacy setting specification task due to its complexity [42, 59]. Thus, a substantial research effort has been done to automatically configure user privacy settings with minimal effort. For instance, [80] shows a comparison of user-defined sharing policies with an automated mechanism based on machine learning. The results show that machine learning approaches have better accuracy than user-defined policies. Semi-supervised and unsupervised approaches have been investigated in [34, 39] to automatically extract privacy settings in social media.

Differently from the above mentioned papers, we use semi-supervised machine learning tools to set up privacy preferences in PDS, by taking into account different features, which are specific of the PDS scenario (e.g., benefits, trust in the third party). Other research work proposed privacy preserving frameworks using machine learning tools [49, 112, 113]. However, in these work the focus is different from our proposal, in that authors primarily focused on privacy-preserving distributed computation, that is, how data holders can collaborate to find predictive models from their joint data without revealing their own data to each other.

2.5 Privacy preferences learning based on contextual data

A number of studies have been carried out to understand user's privacy preferences by take into account user's contextual data. Researchers already implemented contextual based privacy preferences in smart-phone environments. For instance [78, 96, 105, 106] proposed a mechanism to predict permission decisions at runtime that relies on user's contextual information in mobile platforms, whereas [95, 109, 117] proposed user's location sharing privacy preferences by considering contextual information. L. Yuan et al. [116] presented a privacy-aware model for photo sharing based on machine learning that exploits contextual information.

Smith et al. [87] presented different solutions that enable people to share contextual information in mobile networks, whereas, in [104], authors observed that people's willingness for sharing information are impacted of various factors. In [45] presented a framework for automatic estimation of privacy risk of data based on the sharing context. T. liang et al. [56] developed a learning approach that recommends context-aware app by utilizing a tensor-based framework so as to effectively integrate user's preferences, app category information and multi-view features. In [94], authors presented a privacy preference model for helping users to manage their privacy in context-aware systems in term of sharing location on the basis of the general user population using crowd-sourcing architecture. Bigwood et al. [12] have evaluated different machine-learning algorithms so as to build information sharing models. Schlegel et al. [81] proposed a mechanism that summaries the number of requests made by the requesters for sharing his/her location. Liu et al. [60] proposed a decision making approach for sharing private data (such as location) in the context of Android permissions. With this aim, they have built each user profile based on their decision and make users' clusters based on their profiles similarities and then build a single predictive model for each cluster of users. Bilogrevic et al. [14] presented a privacy preference framework that (semi-)automatically predicts sharing decision, based on personal and contextual features. The authors only try to focus on general information sharing with nearby people such as location. These approaches are not able to set up privacy preferences by considering others contextual data such as time, activities etc. However, our proposal having learning model that exploit all possible user's contextual data for setting privacy preferences in PDS.

2.6 Limitations in the existing privacy preference mechanisms

Most of the existing techniques offer users to set up their privacy preferences manually by selecting yes/no options for a set of questions. Interestingly, several studies have shown that with this approach, average users might have difficulties in properly setting potentially complex privacy preferences [3, 59]. To resolve this complexity, we proposed privacy preference frameworks that exploit machine learning tools for learning user privacy preferences according to user feedbacks so as to set up privacy preferences automatically. Some recent research work also exploit machine learning tools to implement privacy preference frameworks [14, 75, 80]. The main intension of these studies is to design a framework that can automatically configure user privacy preferences with minimal user efforts. But the fact is that they did not investigate how the correlations of the learning features can impact the user decisions in term of configuring privacy preferences in PDS. Considering this issue, we extensively exploit different machine learning approaches to check how the correlations of various dimensions of features impact user decisions and improve the performance of the proposed framework. With this motivation, we use single-view, multi-view and ensemble learning techniques. Our experiments show that ensemble approach provides better result over other approaches. Consequently, it confirms that user decisions on access requests are influenced by the correlation of the learning features. Moreover, we also exploit history-based active learning with ensemble approach so as to reduce user efforts to get good quality labeled training dataset that can improve the prediction accuracy w.r.t user's decisions on access requests. The benefit of this framework is in a better understanding of the requirements needed for configuring users' privacy preferences in PDS. Moreover, this results in a better choice and usage of machine learning techniques so as to design more effective and efficient privacy preference frameworks.

2.7 Chapter summary

In this chapter, we focused a privacy preference frameworks to PDS owners for making PDS owners able to configure PDS privacy preferences according to their requirements. For privacy preference, different frameworks have been discussed. These methods mainly focus on privacy specification and enforcement for ensuring user privacy in different domains. Privacy preference in PDS is a relatively new issue and has not been deeply studied. The most primitive and important proposal for configuring privacy preference in PDS is based on answers/questions that can breach user privacy by sequence similar kinds of questions. To address this problem, some proposals have been discussed based on machine learning. In the following of this thesis, we will analyze more in depth the privacy preference by proposing new frameworks based on machine learning so as to configure user privacy preference automatically. We will also focus on the usage of different kind of machine learning approaches with little bit of modifications for trying to find the best privacy preference framework.

Chapter 3

A Risk-Benefit Driven Architecture

3.1 Introduction

In recent years, research in the field of PDS has grown drastically. This has resulted into the definition of several techniques to protect personal data so that unauthorized third parties cannot get access to PDS [6, 35]. Although, such papers proposed some privacy policies to protect personal data from unauthorized access, nevertheless they did not consider the fine-grained holistic view of data owners' perspective regarding their data.

We strongly believe that the decision on whether to release or not personal data is a subjective matter and depends on the evaluation of the benefits and risks connected to the data release. Thus, in this chapter we propose a risk-benefit approach for the design of a privacy-aware PDS. Data owners evaluate access requests from two different perspectives. One is related to the risk of sharing the requested data with third parties, whereas the other one is related to the benefits the data owner can get by accepting the access request for getting the online service. Thus, different data owners may have different evaluation about the same risks and benefits of an access request.

We also believe that the level of sensitivity of personal data in PDS is not equal and also depends on the aptitude of the PDS owner. Thus, we consider that individual may want to give more privacy protection on confidential data to preserve them in more secure ways than non-confidential data. The sensitivity level of data is a subjective matter. For example, user A may consider that his/her email address should not be publicly disclosed. So in this case, (s)he wants to assign a high sensitivity to such data. On the other hand, user B may consider that his/her email address does not need to be hidden. Another dimension that should be taken into account when processing an access request is whether or not the requested data are really *indispensable* to get the service. For instance, a third party[1] like eBay, which is a e-commerce company, sales its products via Internet. Thus, it generates an access request to access PDS, including data fields like credit card number

[1] We use the terms data consumer, third party, or online services interchangeably.

for payment purpose and home address for delivery purpose. However, all the data fields mentioned in access request may not be indispensable for providing the service.

Unfortunately, most of the available approaches do not take into account data confidentially and indispensability when designing the architecture of the PDS. To overcome this, in this chapter we confined PDS data into two data zones. One is confidential data zone and another one is non-confidential data zone. As it will be discussed in the chapter, data stored in the confidential zone will be released only if: (1) they are indispensable by the requesting service, and (2) the data consumer privacy practice satisfies the user preferences. This strategy implies that in case the requested data is confidential but non-indispensable by the requesting service, the access is automatically denied, without user preferences evaluation. This brings benefits in access evaluation efficiency, but also it lows the risk of evaluating mis-configured privacy preferences, specified by not expert users, on confidential data.

In addition to this, the proposed PDS is equipped with a variety of strategies to evaluate an access request on non-confidential data that depends on the satisfaction of user privacy preferences, as well as on the data indispensability and the confidentiality. Such strategies are able to trade between benefits and risks of the data release. Therefore, this chapter provides the following contributions:

- To the best of our knowledge, we are the first proposing a new model for privacy-aware PDS design that takes in consideration the risks and benefits of data release with respect to the data owner's perspective.
- We propose a new architecture for privacy-aware PDS, where we consider two logical data zones: confidential and non-confidential data zone that impact the strategies to perform access control.
- We devise a set of strategies for the release of personal data able to take into consideration various dimensions of the data release, i.e., privacy, efficiency, and the risk-benefit trade-off.

The rest of this chapter is organized as follows. Section 3.2 introduces the overall concepts of our proposed privacy-aware PDS's architecture. Section 3.3 presents a summary of privacy preferences for PDS. Section 3.4 describes the enforcement strategies, whereas, Section 3.5 illustrates the requirement to calculate the risk and benefit values.

3.2 Privacy-aware PDS

The typical PDS architecture does not consider different data zones based on the data sensitivity level. To cope with this issue, we impose a logical separation between confidential and non-confidential data so that we can take extra care on confidential data to protect them from third parties.

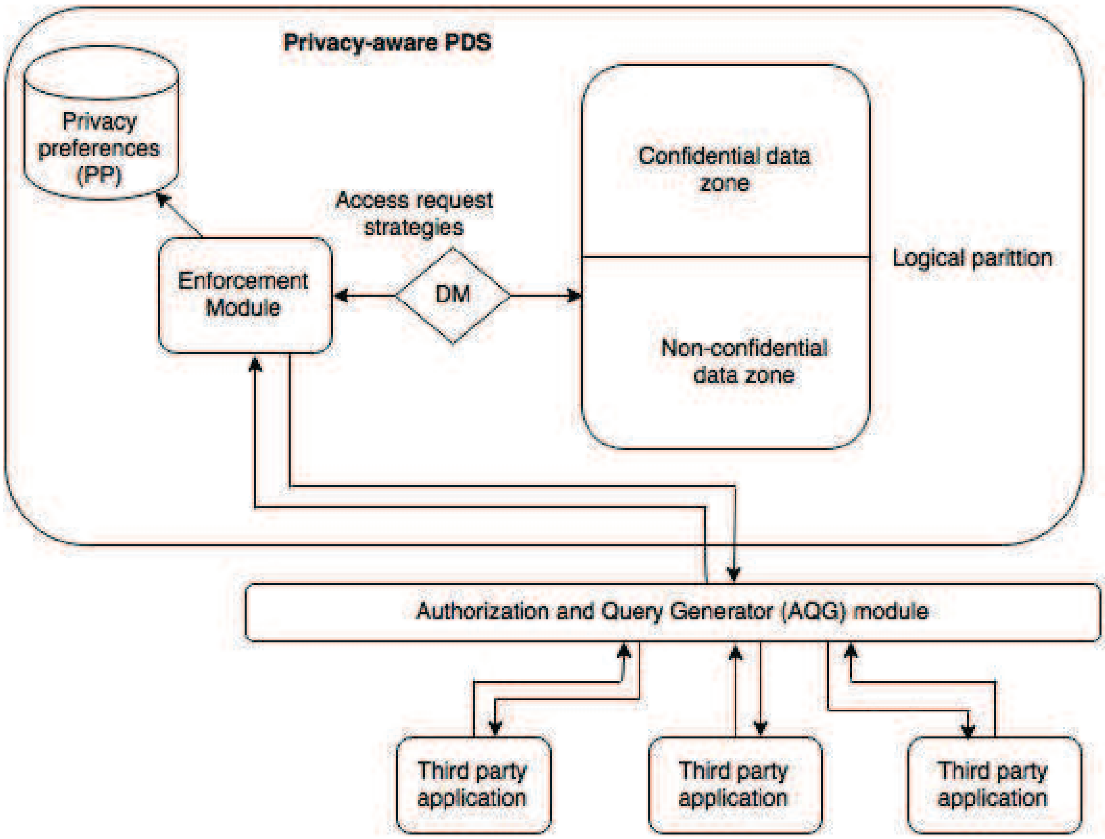


Figure 3.1: Privacy-aware PDS architecture

The proposed architecture is shown in the Figure 3.1. In our model, we divide the Personal Data Storage (PDS) logically in two different data zones: confidential and non-confidential data zone. Basically, we consider that data owners can assign a sensitivity level to their personal data. We also consider that data consumers can assign levels to the requested data objects with respect to indispensability for providing services. Thus, the requested data are partitioned into indispensable and non-indispensable in terms of data consumers' perspective. For each data zone, we propose several strategies for data release. Our idea is that the external application makes request to access PDS data through the Authorization and Query Generator (AQG) module. The AQG module allows only the registered data consumers to access PDS. AQG generates queries based on the access request to obtain the requested data objects that the external applications need for providing services to the data owners.

Then the Enforcement Module (EnM) verifies the queries based on predefined rules, aka user privacy preferences, specified on the requested data. First, the EnM checks the categories of the requested data objects. In particular, the following scenarios are possible (see Table 3.1):

Table 3.1: Enforcement Strategies

Data owner's perspective	Data consumer's perspective	Enforcement strategy
Confidential	Indispensable	Measure the weight between benefits and risk
Non-confidential	Non-indispensable	Depend on user benefits
Confidential	Non-indispensable	No permission to share
Non-confidential	Indispensable	Permission to share

- If the requested data objects are *confidential* and *non-indispensable*, then, according to our strategies, the EnM denies the access request, otherwise, EnM enforces privacy preferences on the requested data objects. If the privacy practices adopted by the data consumer sending the access request satisfy all conditions specified in the data owner privacy preferences, EnM sends a request to Decision Maker (DM) to access PDS data, otherwise the access will be denied.

- If the requested data fields are *non-confidential* and *indispensable*, DM shares these data fields, since the data owner assigns them a low sensitivity degree. In this way, our privacy-aware PDS can enhance its performance compared with other PDSs, because typically PDSs consider same level of privacy preferences for all kinds of personal data.

- If the requested data fields are *non-confidential* and *non-indispensable*, DM measures only the benefits. Since the data have been classified by the data owner as non-confidential, we assume that the owner does not see any significant risk connected to their release. If the benefits are higher than zero, then DM shares these requested data fields, otherwise it denies the access.

- If the requested data fields are *confidential* and *indispensable*, DM measures risks and benefits of data release. After measuring these two values, DM compares risks and benefits. If the benefits are higher than risks, then it approves the release of the requested data. If the benefits are less than risks, then it denies the access request; otherwise, it asks the data owner for decision.

When a request is approved with the requested data fields then DM module sends back the data to AQG module. Finally, AQG sends all the approved data fields to the external applications.

Risks and benefits are the key factors of our privacy-aware PDS model. Risk means something for which the users do not want to share their data for getting services from data consumers. On the other hand, benefits mean something for which the users want to take services from data consumers. We will discuss this further in Section 3.5.

A flowchart describing the overall privacy protection strategies of the proposed privacy-aware PDS is shown in Figure 3.2.

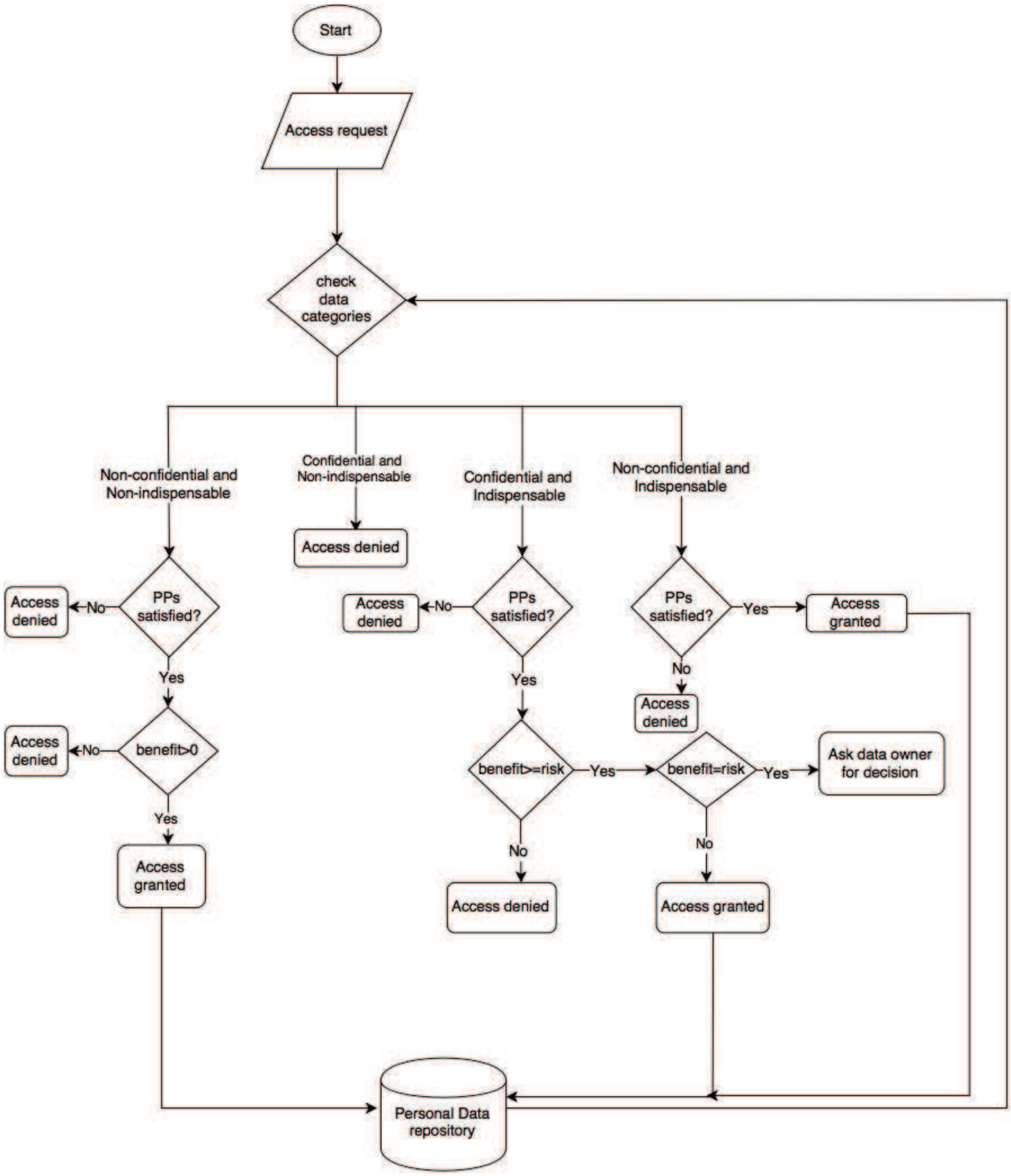


Figure 3.2: An overall flowchart of enforcement strategies

3.3 Privacy Preferences

According the proposed model, privacy preferences are defined based on well-known privacy concepts, like data consumers, data, purposes, services, obligations, conditions, and retention, etc,. In the following, we will describe all these elements in more details.

Data Consumers: We refer to external applications requiring data from PDS as *Data Consumers*. Typically, data consumers are any third-party application. We distinguish data consumers based on their service types and scope, such as commercial data consumers, social media data consumers, medical data consumers, and so on. For example, let assume a PDS owner wishes to purchase items from Amazon site. In this case, he/she needs to allow Amazon as a data consumer to access his/her PDS to share personal data according to his/her privacy preferences for getting this service.

Data: Data elements identify the *data objects* to which a user's preference applies. A PDS may contain different types of personal data, like profile information (Name, Middle Name, Surname, Address, Phone No, E-mail, Birth Date, Gender, Political View, etc.,), credit card information (Card Type, Card Number, Expiry Date, etc.,), eHeath-care information (Blood Pressure, Heart Beat, Weight, Disease Name, Diagnosis Report, etc.,). In our model, we assume data are classified according to two dimensions:

- **Indispensability:** During the registration to PDS, a data consumer has to specify, for each of its services, which data object is indispensable for the execution of the provided service. All the remaining data are considered as non-indispensable for that particular service. For example, suppose that the data consumer is Amazon and the service is book purchasing. Amazon may specify that the data objects mail address for shipment, and credit card information for paying the cost as indispensable data. Moreover, Amazon may also specify some other data objects which are not necessary for providing the service, like phone number and email , but that it requests as well for better service customization. We refer to this type of data objects as non-indispensable data for this particular service.

- **Confidentiality:** Our model assumes that data are classified according to their sensitivity/confidentiality with respect to data owner's point of view. Thus, we assume that each data owner assigns a sensitivity level to his/her personal data.

Purposes: In general, a data consumer notifies the data owner about the intended purposes via privacy policies [24]. Data owners can constraint the release of their data by posing conditions on allowed purposes in the privacy preferences. For example, the privacy statement "the doctor can use the user's health related data to give proper treatment or can use these data for analyzing the reason for which the disease outbreaks in a area" defines two separate purposes, that is, "treatment" and "research" for which the health related data will be used. In this case, perhaps, the data owner wants to disclose his/her data for "treatment" but not for "research". It usually happens that the same data may be used for different purposes. So, the data owner should have the right to state for what purposes the data can be used [58].

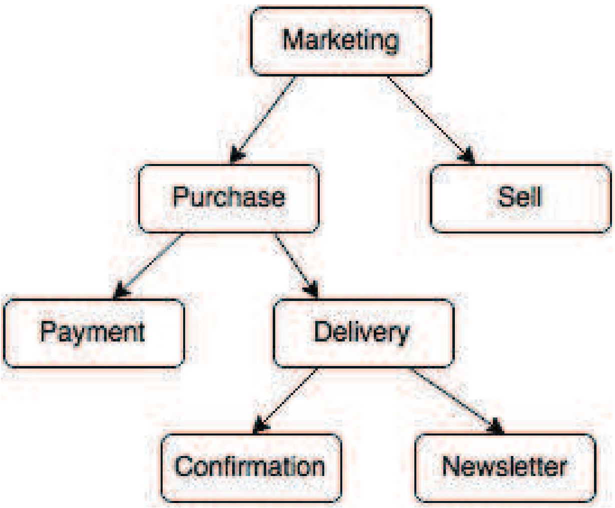


Figure 3.3: A simple purpose tree

In our model, we consider the concept of purpose tree [24] for structuring purposes. Figure 3.3 gives a simple example of purpose tree. Let p be a purpose in purpose tree PT. We consider that descendants (p_i) represent all purposes, including itself p_i which are similar otherwise, we assume that purposes are dissimilar. For example, suppose an access request specifies the marketing purpose, if our model allows this intended purpose to access PDS then it also allows that access for all purposes which are descendants of marketing.

Service: In our view, a service is a process owned by a data consumer through which the data owner has some sort of interaction. For example, eBay provides e-commerce services to a PDS owner.

Obligations: Obligation is a process by which data owners can impose constraints on how personal data should be handled. Every obligation can be associated with an action [10]. Obligations can be divided into two types: pre-obligations and post-obligations. Pre-obligation means fulfilling some conditions before an access is allowed, whereas post-obligations mean fulfilling some conditions after the access is granted. For example, a patient may disclose his/her medical information to doctors for the purpose of treatment and impose that (s)he can not disclose these information to other doctors for the same purpose; this is an example of post obligations. On the other hand, an example of pre-obligation is when doctors need to get legal guardian authorizations to collect medical information of minors for the purpose of treatment.

Context conditions: Privacy preferences can not only specify which data can be used for which purposes but also impose some conditions, besides obligations, that shall be satisfied to get access to data stored into PDS. Such conditions depend on the context in which the request has been issues. For example, a data owner may impose a condition stating that any external application can access his/her PDS only in a particular time frame. A formal definition of context can be defined as follows [1]:

Definition 3.3.1. Context represents any information that can describe the current situation of an entity. An entity can be a person, place, or object that is considered relevant to the interaction between a user and any type of applications.

Retention: The data owner can specify how long his/her data will be alive for processing. The third party application can store, use, and process the data during the time mentioned by the data owner in the retention parameter.

We are now ready to define privacy preferences:

Definition 3.3.2. PDS privacy preference. A privacy preference (PP) regulates the release of personal data stored at PDS and is formalized as follows:

$$PP{=}(DC,\ d,\ p,\ st,\ O,\ c,\ RE)$$

where st is the type of service offered by data consumer DC, d denotes a piece of data, p denotes a purpose, O is the obligation,[2] c denotes the context condition,[3] whereas RE is the retention.

Example 3.3.1. Let us consider a data owner u that wants to regulate the release of his/her blood pressure data to a doctor, only if the request comes from Italy. Moreover, suppose that u wants to disclose this data only for treatment purpose and (s)he wants that the data will not be retained by the service for more than one month. Additionally, (s)he wants to receive a notification via email once the data is accessed. These privacy requirements can be encoded through the following privacy preference:

$$PP = \begin{cases} DC = doctor, \\ d = blood\_Pressure, \\ p = treatment, \\ st = health\_Service, \\ O = (sendemailtome@me.it, \\ c = (IpAddressCountry = Italy), \\ RE = oneMonth; \end{cases}$$

According to the proposed strategy, user privacy preferences have to be evaluated on data consumer *access request*. As described in the following, we assume that an access request contains information about data consumer privacy practice (e.g., purpose, service type, etc.,). In addition to these elements, we also keep into account that data consumers might temporarily offer special treatments associated to the required service, in terms of discount, coupon, etc. While this information does not impact the privacy preferences

[2] Obligation contains a set of actions representing processes that can be executed on the PDS (e.g., sending email, OS primitives, etc.,).

[3] We assume context condition are expressed as boolean expressions over a set of predefined attributes modeling context properties (e.g., access request time, IP address, etc.,).

evaluation, it might impact the estimation of the benefits that might be gained by granting the access (see Section 3.5). Based on this, an access request is formally defined as follows:

Definition 3.3.3. Access request. An access request AR is a tuple:

$$AR\text{=}(DC,\ d,\ p,\ st,\ o)$$

where st is the type of service owned by data consumer DC, requiring via AR to access data d stored at PDS for the purposes p. If the access is granted, DC will provide an additional benefit, called offer, and measured by o.[4]

Example 3.3.2. Let us consider the case an application owned by doctor requests to access the blood pressure for providing health services for treatment purpose. Moreover, let assume that the application temporarily offer an additional discount of 30% for future services. The corresponding access request is modeled as follows:

$$AR = \begin{cases} DC = doctor, \\ d = blood\_Pressure, \\ p = treatment, \\ st = health\_Service, \\ o = 0,30; \end{cases}$$

[4] In this thesis, we assume that an offer is modeled as a value in [0,1]. As an example, it might represent % of offered discount. More complex offers can be modeled as well.

Algorithm 1 Access request evaluation on a PDS

```
INPUT Let $AR$=$(DC, d, p, st, o)$ be the access request submitted to PDS Let $PP\_base$
be the set of privacy preferences specified by PDS owner OUTPUT Access denied, if the
AR is not authorized, Access granted, otherwise Let $pp$=$(DC, d, p, st, O, c, RE) \in PP\_base$
be the privacy preferences that applies to AR, i.e., AR.d=pp.d ; // We assume that a
single privacy preference applies to an access request. The algorithm can
be easily adapted to the case of multiple privacy preferences.
; // We also assume that the access request is on a single data item. The
algorithm can be easily extended to mutiple data items (see Example 3).
if d is confidential and indispensable then
    if (AR.p ∈ descendant(pp.p)) AND (AR.st = pp.st) AND (checkCondition(pp.c, AR)
     then
        r=risk(AR) b=benefit(AR) if $b > r$ then
            enforce Obligations(pp.O) return Access granted
        else
            if $b < r$ then
                return Access denied
            else
                return Ask data owner's decisions
            end
        end
    else
        return Access denied
    end
if d is non-confidential and non-indispensable then
    if (AR.p ∈ descendant(pp.p)) AND (AR.st = pp.st) AND (checkCondition(pp.c, AR)
     then
        b=benefit(AR) if $b > 0$ then
            enforce Obligations(pp.O) return Access granted
        else
            return Access denied
        end
    else
        return Access denied
    end
if d is non-confidential and indispensable then
    if (AR.p ∈ descendant(pp.p)) AND (AR.st = pp.st) AND (checkCondition(pp.c, AR)
     then
        enforceObligations(pp.O) return Access granted
    else
        return Access denied
    end
if d is confidential and non-indispensable then
    return Access denied
```

3.4 Privacy Preference Enforcement

The overall procedure for the evaluation of an access request $AR = (DC, d, p, st, o)$ submitted to a PDS is described in Algorithm 1. We use the dot notation to refer to specific components within a tuple. Algorithm 1 evaluates the access request AR according to our enforcement strategies, (see Table 3.1) and the privacy preferences stated by the data owner explained in Section 3.3. First, Algorithm 1 checks the categories of the requested data objects. Here, we have four cases, according to the assigned data categories. In the first combination, if the requested data objects are confidential and indispensable, then Algorithm 1 checks the privacy preference PP holding on the requested data. If the access request AR is authorized by PP, then Algorithm 1 measures the risk and benefit values, using $risk()$ and $benefit()$ functions, respectively, otherwise, it denies the access request. After measuring the risk and benefit, Algorithm 1 compares these two values. If benefit is greater than risk, then Algorithm 1 grants the access request AR by imposing the obligations. If benefit is less than risk, then Algorithm 1 denies the access request; otherwise, Algorithm 1 asks the data owner for decision. In this case, the Algorithm 1 sequentially checks the elements of privacy preference for authorization and measures the risk and benefit thus, the time complexity of Algorithm 1 is $\mathcal{O}(n)$ where n is the number of comparison.

The second case is when the requested data objects are non-confidential and non-indispensable. In this case, Algorithm 1 checks the privacy preference that applies to the access request AR. If the access request AR is authorized by the privacy preference, then Algorithm 1 measures only the benefit value, using $benefit()$ function, otherwise, it denies the access request. If benefit is greater than zero, Algorithm 1 grants the access request by imposing the specified obligation, otherwise, it denies the access request AR. In this case, the Algorithm 1 sequentially checks the elements of privacy preference for authorization and measures the benefit value thus, the time complexity of Algorithm 1 is $\mathcal{O}(n)$ where n is the number of comparison.

The third case is when the requested data objects are non-confidential and indispensable. In this case, Algorithm 1 applies the privacy preference $PP = (DC, d, p, st, O, c, RE)$ on the access request AR for authorization. If the access request AR is authorized, then Algorithm 1 grants the access request and impose obligations to access data stored in PDS, otherwise, it denies the access request AR. In this case, the Algorithm 1 sequentially checks the elements of privacy preference for authorization and does not need to measure the risk and benefit value thus, the time complexity of Algorithm 1 is $\mathcal{O}(n)$ where n is the number of comparison.

In the final case, if requested data objects are confidential and non-indispensable, then Algorithm 1 denies the access request. In this case, the Algorithm 1 does not need to checks the elements of privacy preference for authorization thus, the time complexity of Algorithm 1 is constant, $\mathcal{O}(1)$.

Example 3.4.1. Let us consider the following access request:

$$\begin{aligned}(eBay, (name, home\_number,\\ street\_name, postal\_code, country, phone\_number,\\ card\_type, card\_number, expiry\_date, card\_owner,\\ email), (delivery, payment), e-commerce,\\ o = 45\%)\end{aligned}$$

Algorithm 1 checks the categories of the requested data fields and enforces the privacy preferences on this access request. In this example, there are two purposes namely, delivery, and payment. Suppose that Algorithm 1 determines that the set of data fields {name, home_number, street_name, postal_code, country} are indispensable, whereas data {phone_number} is non-indispensable for providing e-commerce service for delivery purpose. Similarly, suppose that for payment purpose, Algorithm 1 determines that the set of data fields {card_type, card_number, expiry_date, card_owner} are indispensable, whereas data {email} is non-indispensable for providing e-commerce services. Now, Algorithm 1 searches the requested data fields to find their sensitivity levels with respect to data owners' perspective. Basically, three scenarios may happen.

Scenario-1: Suppose that Algorithm 1 that determines all requested data fields {name, home_number, street_name, postal_code, country, phone_number, card_type, card_number, expiry_date, card_owner, email} of the access request are in the non-confidential data zone. Then, according to our strategies (cfr. Table 3.1), these data fields can be shared, apart from email and phone_number because these are non-indispensable data fields. In this case, we need to find the benefits using the function $benefit()$ for email and phone_number. Thus, Algorithm 1 measures benefits of data owner for these data. If the benefit is higher than zero, then email and phone_number can be shared with data consumer.

Scenario-2: Suppose that Algorithm 1 determines that all data fields {name, home_number, street_name, postal_code, country, phone_number, card_type, card_number, expiry_date, card_owner, email} of the access request are in the confidential data zone, then according to our strategies, Algorithm 1 measures both benefit and risk values using $benefit()$ and $risk()$ and compare them to decide whether the requested data fields can be shared or not (cfr. Table 3.1). In this case, since email and phone_number are non-indispensable data fields, Algorithm 1 does not allow to share these fields according to our strategies. Thus, we can see here that all requested data fields of an access request are not fully shared.

Scenario-3: Suppose that Algorithm 1 determines that some data fields, say {name, home_number, street_name, postal_code, card_type, card_number, expiry_date, email} of the access request are in the non-confidential zone and the remaining data fields, say { country, phone_number,card_owner} are in the confidential data zone, then since the data fields {name, home_number, street_name, postal_code, card_type, card_number, expiry_date} are indispensable and stored in the non-confidential zone, they can be shared, as mentioned in Scenario-1, apart from email because it is a non-indispensable data field. In this case, Algorithm 1 finds the benefits using the $benefit()$ function. If the benefit is higher than zero, then email can be shared with the data consumer. On the other hand, since the remaining data fields { country, phone_number,card_owner} are in the confidential data zone, then, according to our strategies, Algorithm 1 finds out both benefit and risk values using $benefit()$ and $risk()$ and compare them to decide whether these requested data can be shared or not. For phone_number, it is a non-indispensable data field, so it can not be shared. In this case, the requested data are partially shared.

In the following Section, we will discuss the possible ways to estimate risks and benefits.

3.5 Risk-Benefit Estimation

Let us start to consider the benefit of an access request. This measure should estimate the relevance of an access request AR in terms of advantages that the PDS owner might gain if the request is granted. A first positive consequence of a granted access request is that data consumer will be able to provide the service specified in the AR. This might impact the data owner in several ways. The advantages gained by the provided service really depends on the data owner, and, in particularly, on how much that service is needed by the owner. As an example, benefits provided by heart monitoring services will be greatly felt by a heart patient, more than an healthy person who does not need that service. As such, we believe that benefits of an access request have to be measured in terms of data owner necessity of the service requiring the data. Moreover, as introduced in Section 3.3, data consumer might provide temporal offer that could impact the benefit estimation. This brings to define the benefits from an access request as a function of the offer and service need, as stated in the following.

Definition 3.5.1. Benefit. Given an access request AR, the benefit from AR is estimated by a function taking as input the offer value $AR.o$ and the service necessity of $AR.st$, denoted as $need\_st$:

$$b\_AR = benefit(o, need\_st) \tag{3.1}$$

where, $benefit(o, need\_st)$ is a function which has two variables, namely offer o, and service necessity $need\_st$. If service necessity, $need\_st$ and offer o are increased then the value of the function $benefit(o, need\_st)$ is also increased proportionally. We assume that the value of the function, $benefit(o, need\_st)$ is in the range of 0 to 1.

It is obvious that estimation of service necessity depends on the owner, as different persons have different needs. Therefore, we strongly believe that we need to take into account also data owners' attitude regarding services need. At this purpose, a promising approach we intend to follow is to learn this attitude from the data owner itself. Basically, we can consider two phases for estimating benefits. In the first phase, we can ask for data owners' feedback whether he/she needs the service or not. More precisely, we ask data owners to get necessity judgments for each service type and purpose and use these feedbacks as training data. In the second phase, the learning model learns automatically to estimate service's necessity of the upcoming services from PDS using the collected necessity judgments (training data set). Several machine learning tools (e.g., [21, 120]) are available for this purpose, that we plan to use for predicting the judgment of data owner service necessity, $need\_st$. The other element of $benefit(o, need\_st)$ is offer o which may be expressed by data consumers in their access request. Thus, we can measure benefit of data owner regarding an access request using these two values.

Let us now consider the risk estimation. For this, several approaches can be followed, as literature has deeply investigated the problem of risk estimation. As an example, there are a number of research already accomplished to find out the risk [2, 50, 53, 99] in the context of social-media and access control. When dealing with risk of the release of personal data, we have to consider some interesting aspects. The first is that, similarly to other web service scenarios (e.g., e-commerce, e-business, etc), data consumer reputation might play a relevant role in estimating the risk. Under the assumption that the more trusted is the data consumer, the less is the risk, we can rely on work on trust/reputation estimation to measure the risk. As an example, we could exploit proposal in [83], which measures the trust value of data consumers. In addition to data consumer, another dimension we might consider for risk estimation is given by the data requested by the access request. Definitely, more confidential is the data more risky is the access. For this reason, we see risk estimation as a function depending on data consumer and requested data.

Definition 3.5.2. Risk. Given an access request AR, risk is a function of data consumer $AR.DC$ and the requested data $AR.d$ in the access request:

$$r\_AR = risk(d, DC) \tag{3.2}$$

where, $risk(d, DC)$ is a function of two variables, data confidentiality d, and data consumer's reputation DC. Here, the value of $risk(d, DC)$ is decreased as data consumer's reputation DC is increased, whereas the value of $risk(d, DC)$ is increased as data confidentiality d is also increased. We assume that the value of the function, $risk(d, DC)$ is from 0 to 1.

Example 3.5.1. In Example 3, data consumer eBay provides service e-commerce and submits an access request to data owner u's PDS. In this case, to measure the benefits regarding this access request, we need to concentrate on the necessity judgment of data owner u and the offer value regarding e-commerce service provided by eBay. We assume that our system collects sufficient training data regarding this service type. It means that it gets a sufficient level of feedbacks from data owner u about e-commerce. Then using machine learning techniques, we can predict the value of service necessity $need\_st$ precisely. Finally, using benefit(o, need_st) function, we can measure the benefits for which the data owner u would like to take the service by sharing his/her data fields $\big\{$(name, home_ number, street_ name, postal_ code, country, phone_ number), (card_ type, card_ number, expiry_ date, card_ owner, email)$\big\}$ with data consumer eBay for purposes $\big\{$delivery, payment$\big\}$. On the other hand, the data owner u might be anxious regarding his/her privacy threads for sharing these data fields with the data consumer eBay. We can take the trust value of eBay and the confidentiality of $\big\{$(name, home_ number, street_ name, postal_ code, country, phone_ number), (card_ type, card_ number, expiry_ date, card_ owner, email)$\big\}$ into account to measure risk value. After measuring risk and benefit values, our system compares these two values for taking decision whether the requested data fields are (partially) shared with eBay or not.[5]

3.6 Chapter summary

This chapter presents a model that efficiently organized the personal data into PDS based on data confidentiality for the management of user privacy preferences, allowing data releasing with third parties based on risk-benefit assessment. Since privacy preferences is a user subjective matter, thus we introduced the idea of enforcing the authorization privacy policy by using two metrics: risk and benefit. We measured these two values taking into account different dimensions of data owners' perspectives. Furthermore, we plan to implement a prototype of our privacy-aware PDS and test it in different real world scenarios using machine learning approaches.

[5] Although, according to our strategies, we do not measure the risk and benefit for all kind of requested data fields.

Chapter 4

Privacy learning models for PDS

4.1 Introduction

In this chapter, we make a first step that consider the issue of helping users in protecting their PDS data by showing how machine learning tools, which have been extensively used in the literature for many different learning tasks [49, 112, 113], can be used to learn how to answer to future third party data requests, by exploiting user feedback on a training dataset. In particular, the proposed privacy preference learning approach has been designed by considering different driving dimensions, derived from a typical data access request (e.g., intended purpose of usage, service type requiring the data, data consumer, etc.). In selecting the learning approach, we have considered that these dimensions might have a different weight in the decision process of each single individual. More precisely, we wanted to explore if correlations among dimensions impact the decision process of individuals in releasing their data. As such, we have experimentally tested different learning algorithms, namely single-view [23], multi-view [111], and ensemble learning [79], to see which one fits better in our scenario. Moreover, we carried out several experiments to evaluate the effectiveness of the proposed learning approaches in capturing users' preferences regarding privacy-preserving data release. At this purpose, we developed a web application through which evaluators were able to: (1) label a training dataset of access requests to PDS data, and (2) give their feedback on access decisions (i.e., grant/deny) suggested by the system for new access requests. We have tested the developed techniques through two different groups of evaluators. First, we considered a group of students from two universities from Italy and Bangladesh. Then, we used a crowd-sourcing platform to have a bigger and more heterogeneous dataset. The obtained results show that around 79.08% of university based evaluators and 79.24% of crowd-sourcing based evaluators are satisfied with the decisions suggested by the system. As discussed in Section 4.4, the ensemble learning approach returns the best accuracy, which confirms that decision process of individuals in releasing their data takes in consideration correlations among dimensions rather than their simple values.

As discussed in Chapter 2, Section 2.4, recently a learning approach for the suggestion of personalized privacy settings in a similar context has been proposed [75]. However, to the best of our knowledge, we are the first showing that correlations among dimensions of an access request impact individual decisions process.

The rest of this chapter is organized as follows. Section 4.2 gives an overview of our proposal. Section 4.3 presents a summary of the machine learning approaches we use, whereas, Section 4.4 illustrates the experimental results.

4.2 Overall architecture

Our goal is to design a privacy-aware PDS able to automatically answer to service provider requests, by, at the same time, enforcing PDS owner preferences on how personal data have to be released and used. To achieve this goal, we exploit machine learning to build classifiers to predict whether a new access request has to be granted or denied. The literature offers several learning algorithms that we can use [23] to this purpose. In selecting the best fitting one, we have to take into account all the dimensions characterizing the PDS scenario. The first relevant aspect is that data owner's aptitude towards privacy, i.e., how his/her personal data should be processed and managed, is subjective by nature. Indeed, it is matter of fact that different individuals might have different privacy preferences w.r.t. the usage of their data. To cope with this subjective aspect, we cast our attention on learning algorithms that take into account user feedback (i.e., *supervised learning* approaches). These feedback are in general used to create a training dataset on which algorithms build the classifiers. In our setting, learning is based on access requests (see Figure 6.2). The learned models are then used to answer future access requests generated by service provider, using the automatic evaluation module.

We model an access request in such a way that it conveys the most important information that let individuals take conscious privacy-aware decisions on whether they want to release their personal data to the requesting party as defined in 3.3.3. According to the definition, besides the requested data and the access purposes, which state the reason why data are requested, an access request contains the type of the requesting services (e.g., medical, social, bank services) and an indication of the benefits the user can achieve by releasing his/her data. Indeed, already in the 90's Alan Westin showed in his/her study [102] that the majority of people can be regarded as privacy pragmatics, that is, individuals who weigh potential pros and cons case by case and then decide whether to share information or not. We model benefits in terms of temporal offers.

The training dataset is therefore created asking to PDS owners to express a judgment (label) for each access request in the set. This judgment represents his/her personal opinion on whether the access request should be granted, denied, or whether no decision can be taken (i.e., corresponding to labels: yes, no, maybe). The set of access requests AR together with the assigned labels, LAR, form the labeled training set. However, requiring a PDS owner to label huge amount of access requests would negatively impact the usability and acceptance of our solution. Thus, to reduce the owner efforts, we focus on *semi-supervised learning* algorithms, which have the advantage of achieving a good accuracy

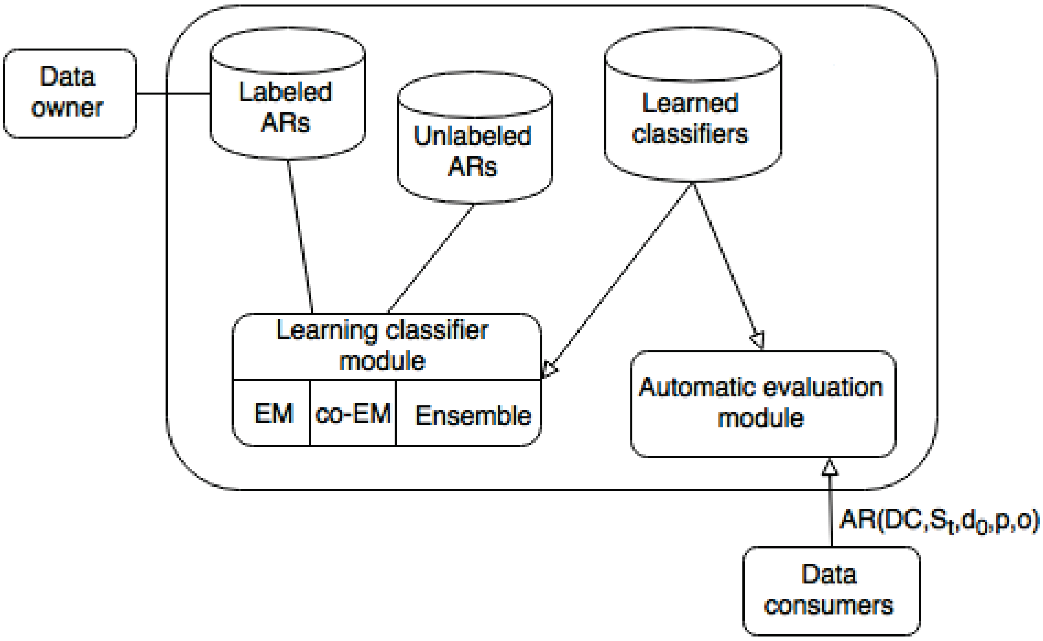


Figure 4.1: Privacy-aware PDS

with a small labeled training set [29, 119]. Semi-supervised learning leverages on both labeled and unlabeled data for the learning tasks [115]. The details of the learning process are given in the next section.

4.3 Learning approaches

Among the available learning approaches, we decided to consider semi-supervised algorithms so as to reduce the user burden on getting training dataset. In Section 4.4, we show that these algorithms provide a better accuracy compared to supervised learning (i.e., SVM) even with a small training dataset. Moreover, in choosing the best semi-supervised algorithm(s) for our scenario, we have kept into account that we expect that individuals give different relevance to each field of the AR (e.g., data consumer, purpose, requested data, etc.). As an example, a user might be more conservative in his/her decisions when dealing with high-sensitive information (e.g., credit card). It is also reasonable to expect that user decisions might be impacted by a combination of different AR's fields. As an example, users with conservative decisions for high-sensitive information (e.g., credit card) might be more prone to data release if data consumers have an high reputation or the returned benefits are relevant (e.g., benefits in terms of service type and/or temporal offer).

In order to keep all these possibilities into account, in this chapter, we focus on three semi-supervised algorithms, namely *single-view*, *multi-view*, and *ensemble*. As it will be described in what follows, the first approach considers the AR as it is, thus building the classifier on the whole set of AR fields all together. The single-view approach has the advantage of being efficient in terms of the time needed for building the classifier, however, it does not exploit possible correlations that may exist among the *AR* fields. For this reason, we test two additional approaches. In the multi-view approach, we consider two disjoint views of AR fields (i.e., a view on AR fields about the requested data and one containing AR fields describing how data are used), and we build two classifiers. In contrast, the aim of the last approach is to consider each AR field separately and build a classifier for each of them. The final prediction is then taken by combining predictions of each single classifier.

In proposing these three approaches, we leverage on semi-supervised learning with generative models [119]. The basic idea of this approach is to estimate the distribution of the probability of items (i.e., access requests) to belong to each class (e.g., yes, no, maybe). Indeed, since in our scenario classes of labeled access requests are created by data owners according to their privacy preferences, we expect to have well clustered data, that is, to have probability distributions with well defined forms. Under this assumption, it has been shown that generative models perform better than other semi-supervised learning approaches [119]. As we describe in the following, among generative models we select the Expectation-Maximization (EM) algorithm [19].

In the following, we first provide details on single-view (aka EM) approach, then we will discuss the multi-view approach (i.e., co-EM algorithm [76]) and, finally, the ensemble approach [36, 88].

4.3.1 Single-view: EM Algorithm

According to this approach, all item features are used together in the learning procedure. This implies considering each access request as a single item. As described above, we exploit semi-supervised learning with generative models, where each probability distribution is parametrized by a vector Θ. In particular, we exploit the EM algorithm, which iteratively estimates Θ for each distribution, so as to optimize the maximum likelihood [53]. To this purpose, the EM algorithm alternates two steps: an expectation (E) step, where it computes the maximum likelihood estimation for Θ, quantified by the log-likelihood of all the items, based on the current estimation for Θ parameters; and a maximization (M) step, where the algorithm updates Θ to maximize the likelihood. Due to space limitations, we do not provide details on the statistical model behind EM, by referring the interested reader to [13, 21].

As depicted in Algorithm 2, in the first iteration the EM algorithm assigns random positive values to Θ parameters (line 4) and evaluates the initial value of the log likelihood $L(\Theta)$ (line 5). In the expectation step, it calculates the membership probability for each item (i.e., access request) in each class (i.e., yes, no, maybe). The algorithm then assigns to each class a set of access requests that, based on their feature values (i.e., access request fields), most likely belong to that class (i.e., that most likely have to be denied, granted or no decision can be taken) (line 7). Then, for each class, parameters are optimized based on the current membership probability in the maximization step (line 8). The expectation and maximization phases are iterated until log-likelihood does not change (i.e., the difference

Algorithm 2 EM(*dataset*)

```
Input: $dataset$, a set of items;
Output: $\Theta$ parameters values;
Let $convergence$ be initialized to false
Let $\Theta$ be initialized with random positive numbers
Calculate the value of $L(\Theta)_{old}$
while $convergence$==false do
  E-Step: Estimate likelihood on $dataset$ based on $\Theta$ parameters values
  M-Step: update $\Theta$ parameters values
  Calculate log-likelihood $L(\Theta)$
  if $|L(\Theta)_{old}$ - $L(\Theta)| \leqslant \epsilon)$ then
    $convergence$=true
  else
    $L(\Theta)_{old}$=$L(\Theta)$
  end if
end while
Return: $\Theta$ parameters values
```

between values of log-likelihood computed in two consequent iterations is less than a value ϵ).

Algorithm 3 applies EM to our context. Note that, applying EM for semi-supervised learning implies to first execute EM on labeled data to estimate the Θ parameters using only labeled data, i.e., LAR (see line 4). The obtained Θ parameters are then updated and used as values for another EM execution over the whole dataset consisting of labeled and unlabeled UAR data (lines 6 - 16).

Once the semi-supervised EM algorithm has been trained, the returned Θ parameters values are used to classify each new access request AR' submitted to PDS. As presented in Algorithm 4, the label returned for AR' is the one associated with the highest membership probability.

4.3.2 Multi-view (Co-EM) learning approach

Multi-view splits item features in the training dataset into two disjoint views, so as to build a distinct classifier on each view [111].[1] Each classifier is used to classify the unlabeled data, which are then used to *co-train* the other classifier. That is, each classifier is retrained again using, in addition to the initial training set, also some items labeled by the other classifier. Co-training brings benefits in terms of a more efficient usage of unlabeled items and an increased accuracy of the resulting classifiers [11, 16].

To apply multi-view in our scenario, we split AR fields into two disjoint views. The first view, named $view_1$, contains only the data specified in the access request (e.g, d_0

[1] It has been shown that the approach works well even if the two views are not disjoint [22].

Algorithm 3 Single_view(LAR,UAR)

```
Input: $LAR$, the set of labeled access requests; $UAR$, the set of unlabeled access requests;
Output: $\Theta$ parameters values;
Let $convergence$ be initialized to false
$\Theta$=EM($LAR$)
Calculate $L(\Theta)_{old}$
Let $DataSet$= $LAR \cup UAR$
while $convergence$==false do
    E-Step: Estimate likehood on $Dataset$ based on $\Theta$ parameter values
    M-Step: update $\Theta$ parameter values
    Calculate log-likelihood $L(\Theta)$
    if $|L(\Theta)_{old}$ - $L(\Theta)| \leqslant \epsilon)$ then
        $convergence$=true
    else
        $L(\Theta)_{old}$=$L(\Theta)$
    end if
end while
Return: $\Theta$ parameters values
```

Algorithm 4 Evaluate_Single_view(AR',Θ)

```
Input: $AR'$, a new access request; $\Theta$, the parameters returned by EM($LAR$,$UAR$);
Output: label for $AR'$;
Let $P_{AR'}(Yes)$ be the probability for $AR'$ of being labeled as YES, computed using $\Theta$
Let $P_{AR'}(No)$ be the probability for $AR'$ of being labeled as NO, computed using $\Theta$
Let $P_{AR'}(Maybe)$ be the probability for $AR'$ of being labeled as Maybe, computed using $\Theta$
Return: the label corresponding to max($P_{AR'}(Yes)$,$P_{AR'}(No)$,$P_{AR'}(Maybe)$)
```

field, see Definition 3.3.3). We define a view with this information only as we believe that data owner's decision on granting/denying an access request is greatly impacted by what the service provider is trying to access. In addition to the required data, another relevant information from the access request is how and by whom the data is used. As such, we create a second view, named $view_2$, with the remaining access request fields, that is, the name of data consumer DC, the service type s_t, the purpose of usage p, and the offer o, which well describe by whom and how (for which purpose and for which service type) data are used.

Example 4.3.1. Let us consider an access request AR =(eBay, Online Shopping, {name, home number, street number, postal code, state name, credit card number, credit card expiry date, credit card security code, credit card type, country, email}, {delivery, payment}, 45%). According to our strategy, $view_1$=(name, home number, street number, postal code, state name, credit card number, credit card expiry date, credit card security code, credit card type, country, email), whereas $view_2$=(eBay, Online Shopping, {delivery, payment}, 45%).

In this chapter, we use the co-EM (multi-view) semi-supervised learning algorithm [76], which combines the concept of co-training and Expectation-Maximization algorithm. Algorithm 5 describes all the performed steps.

Initially co-EM assigns positive values to Θ parameters to train a classifier on $view_1$ (line 7), using only the labeled dataset (i.e., LAR_{view_1}). Then, a classifier uses the labeled dataset of $view_2$ (i.e., LAR_{view_2}) and labeled items assigned by $view_1$ classifier (i.e., L_{UAR_1}) for reassigning the appropriate label to unlabeled items (line 11). In the expectation step (line 13), the membership probability for each item is calculated for each class (i.e., yes, no, maybe) and then, for each class, the parameters are optimized, based on the current membership probability in the maximization step (line 14), similarly to the single-view approach. Then, the $view_1$ classifier uses the labeled dataset of $view_1$ (i.e., LAR_{view_1}) and labeled items (i.e., L_{UAR_2}) assigned by $view_2$ classifier for reassigning the appropriate label to unlabeled items (line 18). In the expectation step (line 20), the membership probability for each item is calculated for each class, whereas in the maximization step (line 21), the parameters are updated for maximizing the membership probability. The expectation and maximization steps for each view are iteratively executed until log-likelihood does not change.

Finally, a label is assigned to all items (i.e., access requests) by combining the membership probability for each class (i.e., yes, no, maybe) of $view_1$ and $view_2$ classifier.

Once the co-EM algorithm has been trained, the returned Θ_{com} parameters values are used to classify each new access request AR' submitted to PDS. As presented in Algorithm 6, the label returned for AR' is the one associated with the highest membership probability.

4.3.3 Ensemble learning approach

In this approach, we take into account that each element of an access request carries valuable information to data owners to take the final decision on data release. Moreover, we

Algorithm 5 co-EM(LAR,UAR)

```
Input: $LAR$, the set of labeled access requests; $UAR$, the set of unlabeled access requests;
Output: $\Theta_{com}$ parameters values;
Let $convergence$ be initialized to false
Let $LAR_{view_1}$ be the projection of $LAR$ on $d_0$
Let $LAR_{view_2}$ be the projection of $LAR$ on $(DC,p,s_t,o)$
$DataSet = LAR \cup UAR$
$\Theta_{view_1}$=EM($LAR_{view_1}$)
while $convergence$==false do
  Let $L_{UAR_1}$ be the labels assigned to $UAR$ computed using $\Theta_{view_1}$ parameters
  Let $Dataset_{view_2}$= $LAR_{view_2} \cup L_{UAR_1}$
  Let $\Theta_{view_2}$=EM($Dataset_{view_2}$)
  Calculate $L(\Theta_{view_2})_{old}$
  E-Step: Estimate likehood on $Dataset$ based on $\Theta_{view_2}$ parameters values
  M-Step: Update $\Theta_{view_2}$ parameters values
  Calculate log-likelihood $L(\Theta_{view_2})$
  Let $L_{UAR_2}$ be the labels assigned to $UAR$ computed using $\Theta_{view_2}$ parameters
  Let $Dataset_{view_1}$= $LAR_{view_1} \cup L_{UAR_2}$
  Let $\Theta_{view_1}$=EM($Dataset_{view_1}$)
  Calculate $L(\Theta_{view_1})_{old}$
  E-Step: Estimate likehood on $Dataset$ based on $\Theta_{view_1}$ parameters values
  M-Step: Update $\Theta_{view_1}$ parameters values
  Calculate log-likelihood $L(\Theta_{view_1})$
  if $(|L(\Theta_{view_2})_{old}$ - $L(\Theta_{view_2})| \leqslant \epsilon) \wedge (|L(\Theta_{view_1})_{old}$- $L(\Theta_{view_1})| \leqslant \epsilon)$ then
    $convergence$=true
  end if
end while
Let $\Theta_{com}$ be the combination of $\Theta_{view1}$ and $\Theta_{view2}$ parameters values
Return: $\Theta_{com}$
```

Algorithm 6 Multi_view(AR',Θ_{com})

```
Input: $AR'$, a new access request; $\Theta_{com}$, the parameters returned by co-EM($LAR$,$UAR$);
Output: label for $AR'$;
Let $P_{AR'}(Yes)$ be the probability for $AR'$ of being labeled as YES, computed using $\Theta_{com}$
Let $P_{AR'}(No)$ be the probability for $AR'$ of being labeled as NO, computed using $\Theta_{com}$
Let $P_{AR'}(Maybe)$ be the probability for $AR'$ of being labeled as Maybe, computed using $\Theta_{com}$
Return: the label corresponding to max($P_{AR'}(Yes)$,$P_{AR'}(No)$,$P_{AR'}(Maybe)$)
```

cast our attention on the relationships between a given access request field and other access request components. Indeed, as an example we believe it is relevant to learn aptitudes of owners in accepting/denying an access, based on the correlation between the requested data d_0 and the requesting data consumer DC. Also the relationship between requested data and intended purpose might have a relevant role in the decision. To take these connections into account, we build a distinct classifier on the projection of the labeled training set on each pair of fields of an access request (e.g., (d_0, DC), (d_0, p), etc.).

Table 4.1: Relationships among access request fields used to build the classifiers

AR field	**Relationships**
Requested data	(d_0,DC), (d_0,p),(d_0, s_t), (d_0, o)
Intended Purpose	(p,DC),(p,s_t),(p,o)
Service type	(s_t,DC), (s_t,o)
Offer	(o,DC)

Actually, we do not consider all possible correlations among access request fields, but only those represented in Table 4.1, where we avoid repeated correlations.

The obtained classifiers are then combined together to predict labels for new access requests. At this purpose, we exploit ensemble learning [88, 118]. This approach is a learning technique where multiple learning classifiers are combined to solve a particular computational problem [79]. Generally, ensemble learning is used to improve the prediction or minimize the error, by combining the individual decision of each classifier.

Algorithm 7 Ensemble(LAR, UAR)

```
Input: $LAR$, the set of labeled access requests; $UAR$, the set of unlabeled access requests;
Output: $\Theta_{ensemble}$ parameters values;
Let $Rel$ be the set of pairs ($m$,$n$), where $m$ and $n$ are $AR$ fields
for $X \in Rel$ do
    Let $LAR_X$ be the projection of $X$ on $LAR$
    Let $UAR_X$ be the projection of $X$ on $UAR$
    $\Theta_X = EM(LAR_X \cup UAR_X)$
    $\Theta_{ensemble} = \Theta_{ensemble} \cup \Theta_X$
end for
Return: $\Theta_{ensemble}$;
```

As illustrated by Algorithm 7, we exploit EM algorithm to build a classifier for each combination of AR fields. Once the classifiers have been built, given a new access request AR', we identify to which class (label) AR' most likely belongs to by using the bagging method [79]. Bagging is an effective method for ensemble learning, where the final label is assigned by computing the average of membership probabilities returned by the obtained classifiers (see lines 39 - 42 in Algorithm 8).

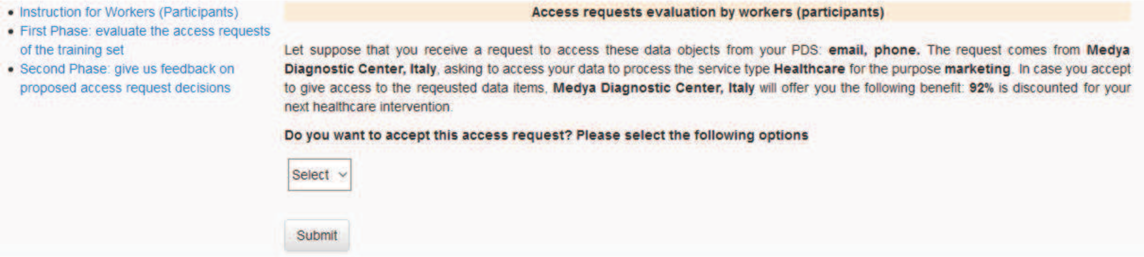
Algorithm 8 Evaluate_Ensemble_view(AR',$\Theta_{ensemble}$)

```
Input: $AR'$, a new access request; $\Theta_{esemble}$, the parameters returned by Ensemble($LAR$,$UAR$);
Output: label for $AR'$;
Let $P_{AR'}(Yes), P_{AR'}(No), P_{AR'}(Maybe)$ be initialized to 0
for $\theta \in \Theta_{esemble}$ do
    Let $P^{\theta}_{AR'}(Yes)$ be the probability for $AR'$ of being labeled as Yes, computed using $\theta$
    Let $P^{\theta}_{AR'}(No)$ be the probability for $AR'$ of being labeled as No, computed using $\theta$
    Let $P^{\theta}_{AR'}(Maybe)$ be the probability for $AR'$ of being labeled as Maybe, computed using $\theta$
    $P_{AR'}(Yes) = P_{AR'}(Yes) + P^{\theta}_{AR'}(Yes)$
    $P_{AR'}(No) = P_{AR'}(No) + P^{\theta}_{AR'}(No)$
    $P_{AR'}(Maybe)$= $P_{AR'}(Maybe) + P^{\theta}_{AR'}(Maybe)$
end for
$P_{AR'}(Yes)$= $P_{AR'}(Yes)/|\Theta_{ensemble}|$
$P_{AR'}(No)$= $P_{AR'}(No)/|\Theta_{ensemble}|$
$P_{AR'}(Maybe)$= $P_{AR'}(Maybe)/|\Theta_{ensemble}|$
Return: The label corresponding to max($P_{AR'}(Yes)$,$P_{AR'}(No)$,$P_{AR'}(Maybe)$)
```

4.4 Experiments

We carried out several experiments to evaluate the effectiveness of the proposed learning approaches in capturing users' preferences regarding a privacy-preserving data release.

To evaluate if a classifier has correctly labeled a new access request, we need to ask a judgment to the user that labeled the training dataset used by the classifier. To this purpose, we set up an experiment that exploits a group of evaluators. In particular, we developed a web application through which evaluators label a training dataset of access requests (i.e., learning phase) and give their feedback on labels associated by the system to new access requests (i.e., the testing phase). More precisely, during the learning phase participants are asked to associate a decision (yes, no, maybe) with 40 access requests. As depicted in Figure 4.2, this is done by posing a question where all access request fields are highlighted. Once this phase has been concluded, for each evaluator, the collected training dataset is elaborated by the learning algorithms illustrated in Section 4.3. Algorithms have been implemented exploiting the R platform [15]. In the testing phase, the web application selects 21 new access requests, generates the corresponding labels using the single-view, multi-view, and ensemble classifiers obtained over the trained dataset. Finally, as shown in Figure 4.3, the web application asks the evaluators whether they are satisfied with the system generated label (i.e., the access decision). The evaluator can answer with yes, no, or maybe.

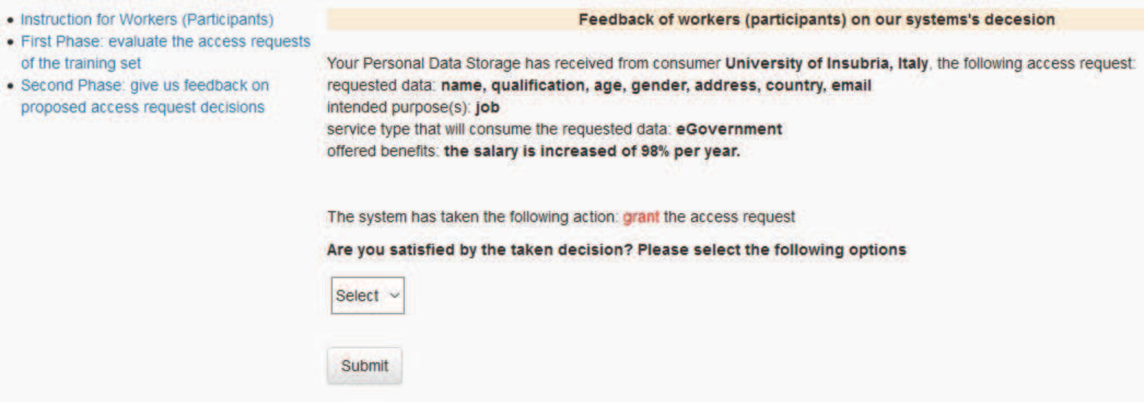

Instruction for Workers (Participants)
First Phase: evaluate the access requests of the training set
Second Phase: give us feedback on proposed access request decisions
Select
Submit

Access requests evaluation by workers (participants)

Let suppose that you receive a request to access these data objects from your PDS: **email, phone.** The request comes from **Medya Diagnostic Center, Italy**, asking to access your data to process the service type **Healthcare** for the purpose **marketing**. In case you accept to give access to the reqeusted data items, **Medya Diagnostic Center, Italy** will offer you the following benefit: **92%** is discounted for your next healthcare intervention.

Do you want to accept this access request? Please select the following options

Figure 4.2: Learning phase


Instruction for Workers (Participants)
First Phase: evaluate the access requests of the training set
Second Phase: give us feedback on proposed access request decisions
Select
Submit

Feedback of workers (participants) on our systems's decesion

Your Personal Data Storage has received from consumer **University of Insubria, Italy**, the following access request:
requested data: **name, qualification, age, gender, address, country, email**
intended purpose(s): **job**
service type that will consume the requested data: **eGovernment**
offered benefits: **the salary is increased of 98% per year.**

The system has taken the following action: grant the access request

Are you satisfied by the taken decision? Please select the following options

Figure 4.3: Testing phase

4.4.1 Settings

Access requests dataset. We consider access requests consist of five elements as defined in 3.3.3, namely, data consumer DC, requested data d, purpose p, service type st, offer value o. In order to generate the dataset of access requests, we have first generated possible values for each access request field. We assume that each possible combination of these values can represent an access request. However, to make our dataset realistic, we have defined 4 service types (i.e., Online Shopping, Banking, Healthcare, eGovernment), 15 purposes (among which: treatment, diagnostic, eTicket airline, money transfer, direct marketing, job etc.), 50 consumers of different types (e.g., eBay, ItaliaRail, Hospitals, etc.), 35 different data types to be required (e.g., name, home number, disease name, date of birth, weight, email, etc.), and measured the offer as a value in the range from 0 to 100%. Using these values, we create new access requests by first randomly select a value for each field and then by checking the consistency of the obtained access requests, in order to remove those field combinations that are not semantically meaningful (e.g., an access request for online shopping service type with treatment purpose). Moreover, we measure how much time a user devotes to give feedback on an access request. If it is less than a desirable time (i.e., user did not read the access request carefully) then the user is removed from the task. The obtained dataset contains 290 access requests.

Evaluator groups. We use two different groups of evaluators for better understanding the performance of our approach. We exploit the same dataset for both the groups of evaluators.

University-based evaluators: we invite undergraduate students studying in computer science for testing the performance of our proposed privacy preference framework. To do so, we invite them by sending email. We consider 30 students: 11 students from the University of Insubria, Italy, and 19 students from Islamic University, Bangladesh. Students participated in the evaluation using the above described web application.

Crowd-sourcing based evaluators: we exploit the Microworker crowd-sourcing platform[2] for the enrollment of additional participants (called workers) of different nationalities, ages, and educational levels. At this purpose, we select only the workers with the best rating according to the Microworker platform, with the results of 220 evaluators. Once the job has been accepted, each worker has been redirected to our web application to conduct both learning and testing phases.

Metrics. In order to measure the effectiveness of the proposed learning approaches, we make use of conventional measures. In particular, since we have classes with 3 labels (yes, no, maybe), we exploit a 3X3 confusion matrix, see Table 4.2, representing the adopted notations. More precisely, column of the matrix represents the predicted value for a class, row represents a possible actual value, and an element identified by row and column specifies the type of error, if any, in labeling an item whose real value is specified in the row with the label corresponding to the column. From the confusion matrix, we define the evaluation metrics given in Table 4.3.

[2] https://www.microworkers.com

Table 4.2: Confusion matrix

	Predicted value: Yes	Predicted value: No	Predicted value: Maybe
Actual value: Yes	$\mathrm{TP}_{\mathrm{Yes}}$	$\mathrm{E}_{\mathrm{Yes,\ No}}$	$\mathrm{E}_{\mathrm{Yes,\ Maybe}}$
Actual value: No	$\mathrm{E}_{\mathrm{No,\ Yes}}$	$\mathrm{TP}_{\mathrm{No}}$	$\mathrm{E}_{\mathrm{No,\ Maybe}}$
Actual value: Maybe	$\mathrm{E}_{\mathrm{Maybe,\ Yes}}$	$\mathrm{E}_{\mathrm{Maybe,\ No}}$	$\mathrm{TP}_{\mathrm{Maybe}}$

Table 4.3: Metrics definition

Accuracy=$\mathrm{TP}_{\mathrm{Yes}}$+$\mathrm{TP}_{\mathrm{No}}$+$\mathrm{TP}_{\mathrm{Maybe}}$/total number of samples
Precision Yes=$\mathrm{TP}_{\mathrm{Yes}}$/ $\mathrm{TP}_{\mathrm{Yes}}$+$\mathrm{E}_{\mathrm{No,\ Yes}}$+$\mathrm{E}_{\mathrm{Maybe,\ Yes}}$
Precision No=$\mathrm{TP}_{\mathrm{No}}$/ $\mathrm{TP}_{\mathrm{No}}$+$\mathrm{E}_{\mathrm{Yes,\ No}}$+$\mathrm{E}_{\mathrm{Maybe,\ No}}$
Precision Maybe=$\mathrm{TP}_{\mathrm{Maybe}}$/ $\mathrm{TP}_{\mathrm{Maybe}}$+$\mathrm{E}_{\mathrm{Yes,\ Maybe}}$+$\mathrm{E}_{\mathrm{No,Maybe}}$
Recall Yes=$\mathrm{TP}_{\mathrm{Yes}}$/ $\mathrm{TP}_{\mathrm{Yes}}$+$\mathrm{E}_{\mathrm{Yes,\ No}}$+$\mathrm{E}_{\mathrm{Yes,\ Maybe}}$
Recall No=$\mathrm{TP}_{\mathrm{No}}$/ $\mathrm{TP}_{\mathrm{No}}$+$\mathrm{E}_{\mathrm{No,\ Yes}}$+$\mathrm{E}_{\mathrm{No,\ Maybe}}$
Recall Maybe=$\mathrm{TP}_{\mathrm{Maybe}}$/ $\mathrm{TP}_{\mathrm{Maybe}}$+$\mathrm{E}_{\mathrm{Maybe,\ Yes}}$+$\mathrm{E}_{\mathrm{Maybe,\ No}}$
$\mathrm{F1}_C$=2* ($\mathrm{Precision}_C$*Recall_C)/($\mathrm{Precision}_C$+Recall_C), where $C \in$\{Yes, Maybe, No\}

4.4.2 Accuracy

In the first experiment, we provide a comparison of accuracy obtained by different classifiers. More precisely, we first make a comparison between a semi-supervised soft-clustering method (i.e., EM) against a hard-clustering approach (i.e., SVM), so as to show that semi-supervised approaches have a good accuracy even with a reduced training set. As second experiment, we measure and compare the accuracy of the proposed classifiers, namely, single-view, multi-view, and ensemble.

Supervised hard-clustering vs. Semi-supervised soft-clustering. As our aim is to reduce the user burden on getting training dataset, in selecting the learning algorithm we focused on semi-supervised EM learning approach. Nevertheless, we check the performance of hard clustering supervised machine learning techniques to make a comparison. At this purpose, we compare the accuracy of EM single-view against SVM. Results are shown in Figure 4.4. The reported results have been computed by training SVM and EM single-view over 20 access requests of the training set and evaluating the resulting classifiers over the remaining 20 access requests.

The results show that when we consider small amount of training dataset, such as 20 access requests, then the performance of single-view EM is degraded compared with the 40 access requests training dataset (see in Figure 4.5). Moreover, the accuracy level of SVM is less than the one of EM single-view, because SVM needs bigger training datasets for predicting accurate results. This motivates us to use soft clustering techniques rather than hard clustering. Based on this values, we considered for further experiments a training

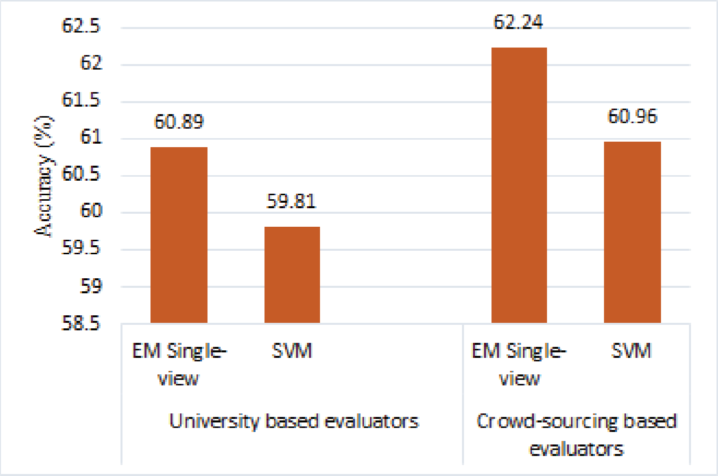


Figure 4.4: Accuracy of 3-classes EM single-view vs. 3-classes SVM single-view

dataset of 40 questions, to trade-off between users burden and good prediction accuracy.

Single-view, Multi-view, and Ensemble accuracy. At this purpose, we use the obtained classifiers to (re)-label access requests in the training dataset LAR, so as to compute the true positive values (i.e., TP_{yes},TP_{No},TP_{Maybe}). As such, we define the accuracy as a ratio of the total number of true positives to the total number of samples. As shown in Figure 4.5, around 74.11% and 75.71% of the training dataset of university based evaluators and crowd-sourcing based evaluators are correctly labeled by the ensemble algorithm, respectively, whereas around 71.07% and 72.27% of the training dataset of university based evaluators and crowd-sourcing-based evaluators are correctly labeled by EM (single-view), whereas 65.71% and 67.69% are correctly labeled by Co-EM(multi-view) algorithm, respectively. Therefore, in this experiment we can see that the performance of the ensemble learning approach is better than single-view and multi-view approaches.

4.4.3 Satisfaction level

This experiment is based on feedback received by evaluators during the testing phase. We recall that in this phase, the web application shows to evaluators a set of access requests with the corresponding decisions (labels), and asks to them if they are satisfied by the taken system decision. In particular, we consider 24 access requests. Among them, 21 access requests have been generated by the classifiers obtained by the three proposed learning approaches (7 access access requests for each approach, randomly presented to the evaluators). The remaining 3 access requests are taken from the set of access requests that the evaluators have labelled during the first phase. These are used for checking the consistency of evaluators judgments, and thus making some considerations on the quality of the evaluators (see the discussion in the next section). As shown in Figure 4.6, around 79.08% of university based evaluators and 79.24% of crowd-sourcing based evaluators are satisfied

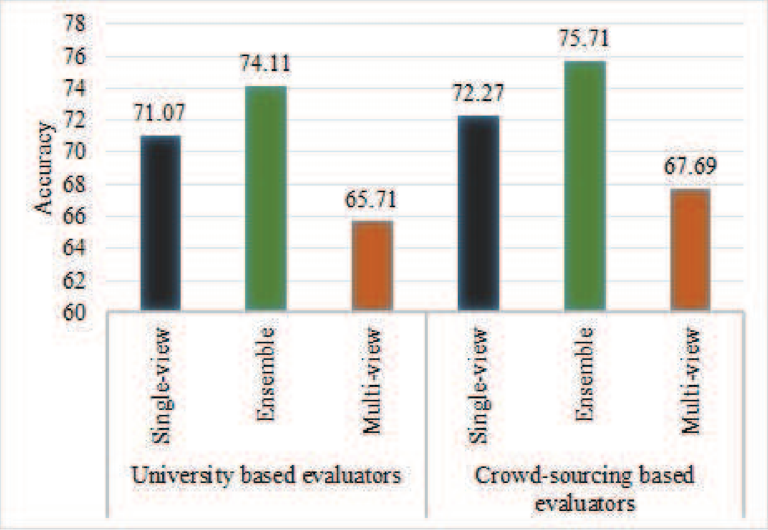


Figure 4.5: Accuracy of single-view, ensemble, and multi-view

with the decisions taken using the ensemble learning classifier, whereas around 74.49% and 75.17% are fine with the decisions taken using single-view classifiers, and 72.45% and 72.25% with decisions taken by using Co-EM. This experiment further confirms that all approaches obtain a good satisfaction level, but again ensemble outperforms the others. This shows that relationships among elements of an access request play a relevant role in the data owner decision of releasing personal data.

4.4.4 Evaluator quality

As the proposed system depends on owner input on the training dataset, we are interested to investigate how a badly labelled training set impacts the final satisfaction level. At this purpose, we make use of the three access requests that, as mentioned earlier, are taken from the set of access requests that evaluators have labelled during the first phase. In particular, in presenting these access requests during the testing phase, the web application shows, as taken decision, the same label entered by the evaluator during the first phase. Based on the satisfaction level the evaluator assigns to this label, we can judge whether the evaluator is consistent or not in his/her decisions, which gives us a measure of the quality of his/her jobs. We consider an evaluator as consistent if he/she is satisfied by the shown taken decisions for all the three access requests.

Figure 4.7 presents the comparative analysis of the satisfaction level for consistent and inconsistent evaluators. The figure shows that the satisfaction level of consistent users is greater than the satisfaction level of inconsistent users. However, even in the worst case, about 68% of inconsistent evaluators are satisfied by the taken decisions.

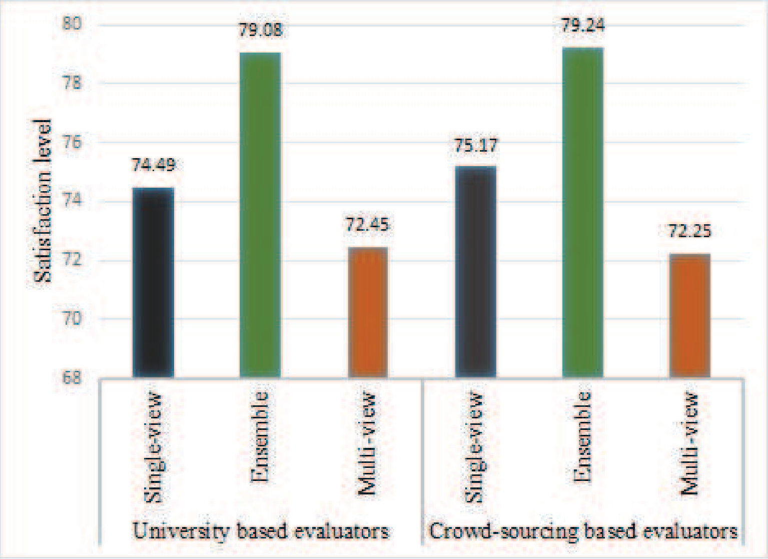


Figure 4.6: Evaluators satisfaction level

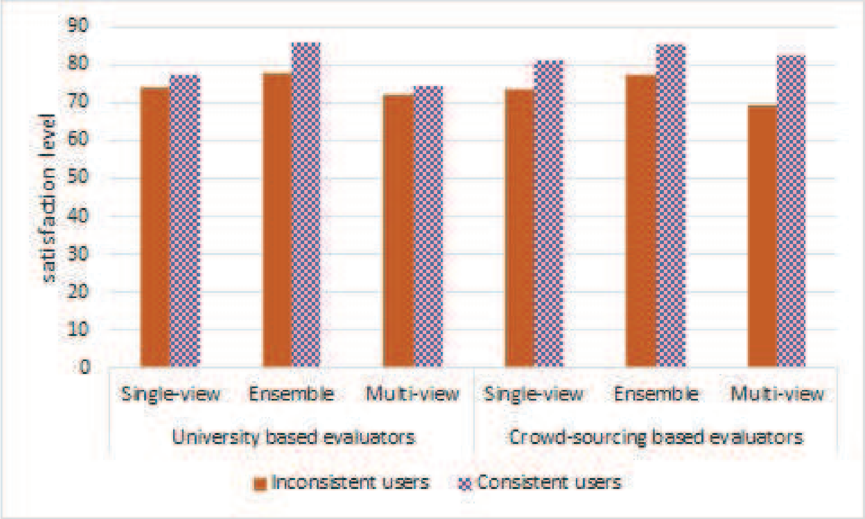


Figure 4.7: Satisfaction level of consistent and inconsistent evaluators

Table 4.4: Comparison of single-view, ensemble, and multi-view for the training dataset

		Single-view			Ensemble			Multi-view		
		Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe
University based evaluators	Precision	77.80 %	66.75%	50.93%	84.92%	67.53%	47.18%	71.53%	63.40%	47.88%
	Recall	77.93 %	71.55%	46.15%	76.14%	75.50%	58.26%	74.56%	65.25%	38.18%
	F1	77.86 %	69.07%	50.16%	80.29%	71.29%	52.14%	73.01%	64.31%	42.63%
Crowd-sourcing based evaluators	Precision	80.33 %	67.75%	52.50%	90.34%	66.08%	42.03%	76.95%	62.04%	45.69%
	Recall	81.58 %	63.89%	54.72%	78.91%	71.02%	66.94%	78.67%	57.27%	48.24%
	F1	80.95 %	65.76%	53.59%	84.24%	68.46%	51.63%	77.80%	59.56%	46.93%

Table 4.5: Comparison of single-view, ensemble, and multi-view for the testing dataset

		Single-view			Ensemble			Multi-view		
		Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe
University based evaluators	Precision	73.95 %	78.85%	68.00%	83.47%	71.67%	73.33%	80.00%	70.83%	61.76%
	Recall	88.89 %	51.25%	100%	85.59%	64.18%	100%	77.42%	62.20%	100%
	F1	80.73 %	62.12%	80.95%	84.52%	67.72%	84.62%	78.69%	66.23%	76.36%
Crowd-sourcing based evaluators	Precision	84.79 %	66.67%	60.42%	81.82%	77.56%	64.63%	80.76%	64.86%	60.63%
	Recall	83.72 %	53.69%	100%	91.51%	53.23%	100%	80.86%	50.62%	100%
	F1	84.25 %	59.48%	75.32%	86.39%	63.13%	78.51%	80.81%	56.86%	75.49%

4.4.5 F1 Measure

We measure the F1 score for each class (yes, no, maybe) in the training dataset and testing dataset, separately (see Tables 4.4 and 4.5, respectively). F1 score considers both the precision and the recall. The precision is the ratio of the number of correctly predicted items (i.e.,access requests) to the total number of incorrectly predicted items. Whereas, recall is the ratio of the number of correctly predicted items to the total number of relevant items. Our analysis exhibits that ensemble learning works better than single-view and multi-view approaches, as shown in Figures 4.8 and 4.9 for training and testing datasets, respectively. From our analysis we observe that in the testing phase, users may agree/disagree on the maybe decision taken by the system, but no one selects the option “maybe” against the decision of yes and no taken by the system. Thus, the recall for “maybe” class is maximum in all the considered approaches. It means that participants are not confused when evaluating the system decisions.

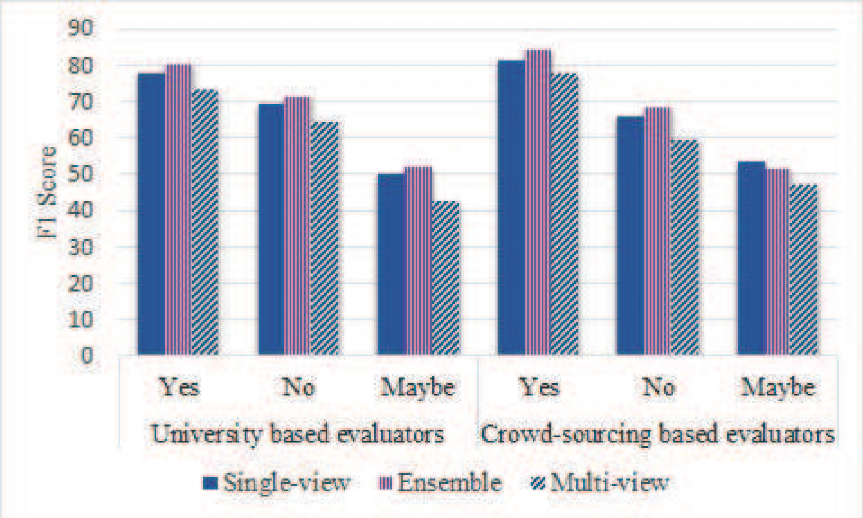


Figure 4.8: Comparison of F1 score for single-view, ensemble, and multi-view for the training dataset

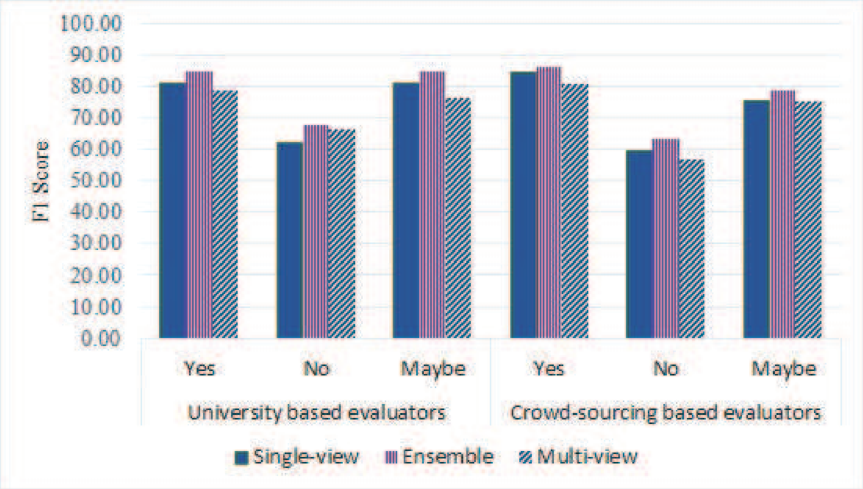


Figure 4.9: Comparison of F1 score for single-view, ensemble, and multi-view for the testing dataset

4.5 Chapter summary

This chapter presented a privacy preference approach that regulating data release from PDS with third party applications. This approach exploits machine learning tools (e.g., single-view, multi-view, and ensemble) that discloses the personal data whose release has been explicitly learned from PDS owners previous feedbacks. We have extensively tested our approaches by evaluators enrolled from the university environment, as well as through a crowd-sourcing platform. The achieved experimental results show that the correlations among the different dimensions of learning elements make the positive impact to learn users' privacy preferences. Thus, ensemble learning approach provides better result over other approaches. The main advantage of our approaches is that they can nicely be interoperable with current solutions in different application domains.

Chapter 5

Privacy learning optimized models for PDS

5.1 Introduction

The privacy preference learning models discussed in the Chapter 4 have used different semi-supervised machine learning approaches for learning privacy preferences of PDS owners. The idea is to find a learning algorithm that, after a training period by the PDS owner, returns a classifier able to automatically decide if access requests submitted by third parties are to be authorized or denied. In Chapter 4, we have shown that, among different semi-supervised learning approaches, the one that better fits the considered scenario is ensemble learning [37, 79]. Even though the identification of the learning approach is an essential step, the design of a *Privacy-aware Personal Data Storage* (P-PDS), that is, a PDS able to automatically take privacy-aware decisions on third parties access requests requires further investigation. One critical aspect to consider is the usability of the system. As several studies have shown, average users might have difficulties in properly setting potentially complex privacy preferences [3, 63]. Even if semi-supervised techniques require less users' effort, compared to manually setting privacy preferences, in order to obtain accurate results they still require many interactions with PDS owners to collect a good training dataset. To reduce the required user effort, in the current chapter, we leverage on active learning (AL) [82] to minimize user burden for getting the training dataset by, at the same time, achieving better accuracy in determining user privacy preferences. The main idea of active learning is to select from the training dataset the most representative instances to be labeled by users. Literature offers several methods driving the selection of these new instances. The most commonly adopted method is *uncertainty sampling* [82]. According to this approach, to be labeled by human annotators, active learning selects those instances for which it is highly uncertain how to label them according to the preliminary built model. As reported in Section 5.5, this improvement brings benefits in term of accuracy and usability. Additionally, to further improve the performance of the system we define an alternative uncertainty sampling strategy, which is based on the observation that, for taking a privacy-related decision, some fields of access requests (i.e., data consumer and type of service requiring the data) are more informative than others. Thus, if a new access request presents new values for these fields, the system push for a new training (i.e., asking data owner a label for the access request). To enforce this behavior, we introduce a penalization of the uncertainty measure based on the *distance* of the new access request w.r.t. the access requests previously labeled by the P-PDS owner (we call this strategy *history-based active learning*). As it will show in the experiments, history-based active learning shows better results than AL in terms of user's satisfaction. As a further improvement, in this chapter, we propose a revised version of the ensemble learning algorithm proposed in the Chapter 4, to enforce a more conservative approach w.r.t. users' privacy. In particular, we reconsider how ensemble learning handles decisions for access requests for which classifiers return conflicting classes. In general, the final decision is taken selecting the class with the highest aggregated probabilities. However, this presents the limit of not considering users perspective, in that, it does not take into account which classifier is more relevant for the considered user. To cope with this issue, we propose an alternative strategy for aggregating the class labels returned by the classifiers. According to this approach, we assign a personalized weight to each single classifier used in ensemble learning. We also show how it is possible to learn these weights from the training dataset, thus without the need of further input from the P-PDS owner. Experiments show that this approach increases user's satisfaction as well as the learning effectiveness.

The rest of this chapter is organized as follows. Section 5.2 gives an overview of our proposal. Sections 5.3 and 5.4 present the proposed learning approaches, whereas Section 5.5 illustrates the experimental results.

5.2 Privacy-aware PDS (P-PDS)

In general, to build a classifier using a predictive learning model, it is essential to label an initial set of instances, called the training dataset. However, it is matter of fact that obtaining a sufficient number of labeled instances is time consuming and costly due to the required human input [31]. On the other hand, the size and quality of the training dataset impact the accuracy the classifier might reach. At this purpose, active learning (AL) [82] may be exploited to reduce the size of the training dataset. The key idea of AL is to build the training dataset by properly selecting a reduced number of instances from unlabeled items, rather than randomly choosing them as done by traditional supervised learning algorithms. This makes it possible to efficiently exploit unlabeled instances for developing effective prediction models as well as to reduce the time and cost of labeling [20].

More precisely, the main idea of active learning is to first select very few instances for being labeled by human and build on them a preliminary prediction model. After that, AL exploits this preliminary model to select new instances from the training dataset to be labeled to reinforce the model. Literature offers several methods driving the selection of these new instances. The most commonly adopted method is *uncertainty sampling* [82]. According to this approach, AL selects those instances for which it is highly uncertain how

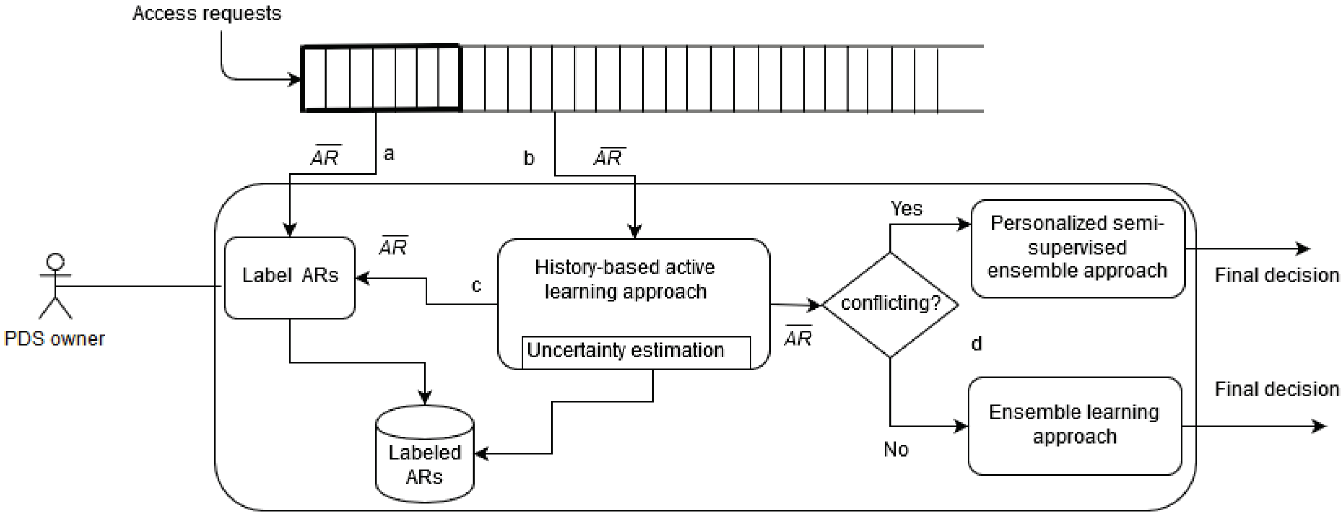


Figure 5.1: P-PDS architecture

to label them according to the preliminary built model, thus that require to be labeled by human annotators.

Although AL greatly reduces human participation on labeling training datasets and leads to good performance, researchers have further investigated how to combine active learning with semi-supervised approaches [67, 72]. We recall that semi-supervised learning algorithms can learn from labeled and unlabeled data, as such AL can improve this approach by properly selecting the most uncertain unlabeled data to be labeled, thus to further reduce the cost of labeling. This nice benefit motivates us to adopt this strategy and to design a *privacy-aware PDS (P-PDS) that deploys the ensemble learning algorithm proposed in Chapter 4 but following an active learning approach* so as to minimize user burden for getting the training dataset and, at the same time, to achieve excellent performance to predict accurate classes for unlabeled data (aka, new access requests submitted to the P-PDS).

As depicted in Figure 6.2, the proposed P-PDS selects a first small set of incoming access requests (Figure 6.2.a) in order to create an initial training dataset, to be labeled by the P-PDS owner, which is then used to build the preliminary learning model. Then, using this preliminary model P-PDS measures the uncertainty of the newly arriving access requests $\overline{AR}$ (Figure 6.2.b) and asks P-PDS owner to directly label $\overline{AR}$ only if its uncertainty level is high (Figure 6.2.c). Otherwise, $\overline{AR}$ is immediately labeled by the semi-supervised ensemble classifier using the preliminary learning model. In this way, AL can minimize user burden by only labeling most uncertain access requests leading a good accuracy and making P-PDS able to take decision even with a small training dataset.

Even if this improvement brings benefits in term of accuracy and usability, we believe that it can be further extended so as to be more conservative w.r.t. P-PDS owner's privacy. This consideration arises from the following example. Let us consider these two access requests: AR'(Amazon, online shopping, mail address, credit card information, delivery and payment, 50%) and AR''(MyAmazon, online shopping, mail address, credit card information, delivery and payment, 50%), which are identical apart from the consumer. Let also assume that AR' has been already labeled by the P-PDS owner. By adopting an AL strategy, the P-PDS might consider AR'' not to be labeled, as the uncertainty value is low since only one field differs. However, in doing so, we would not consider that the consumer field is too informative to not consider its variation. The issue is that AL does not consider the semantics of AR's fields, and their relevance in the P-PDS owner's decision process. Indeed, a user might fully change his/her decision on an access request based on the requesting data consumer (aka its reputation). Thus, we believe that it would be relevant to give extra consideration to access requests coming from new data consumers. In addition to this field, we also believe that service type is a key element with respect to data owners' perspective. In reality, granting/denying an access request deeply depends on the need the individual has for that type of service. For instance, in case of health problems some types of service (e.g., heart-beat monitoring) is not only needed but it is mandatory for individual survival.

For this reason, when an access request comes from a new data consumer or with a new service type, the P-PDS triggers the P-PDS owner for labeling the new request. At this purpose, we decide to complement AL with additional strategies for triggering the selection of new instances to be labeled. More precisely, we revise the strategy of uncertainty sampling, traditionally adopted in AL to increase accuracy, so as to increase the level of uncertainty based on values of data consumer and service type of the newly arrived access request. As described in Section 5.3, this uncertainty adjustment is driven by the distance between the value of data consumer/service type of the new access request and the values of the corresponding elements in access requests already labeled by the P-PDS owner. This solution traces the history of labeled access requests, as such we call this model *history-based active learning* (see Section 5.3 for more details).

The second relevant new feature of P-PDS is related on how ensemble learning handles decisions for access request having conflicting class. In general, in order to provide the final decision for a new access request $\overline{AR}$, ensemble computes the probabilities for each classes (i.e., yes, no, maybe) using the $\Theta_{ensemble}$ classifiers. Then it sums all probabilities associated with a given class and selects, as final decision, the class with the highest probabilities. As such, ensemble does not consider the class semantics, aka whether they are conflicting, but it simply aggregates their probabilities. If this works in some application scenarios, in our context it might represent a problem. For example, let us consider an access request AR receiving the following classes: *yes* for $\Theta_{(s_t,d)}$, *no* for $\Theta_{(s_t,o)}$, *maybe* for $\Theta_{(DC,o)}$, *maybe* for $\Theta_{(p,o)}$, *yes* for $\Theta_{(DC,p)}$ and so on. Suppose that, based on the obtained probabilities, the ensemble approach returns the final class label *yes* for AR, even though the decisions produced by the classifiers $\Theta_{ensemble}$ are conflicting. However, this decision might not reflect the correct opinion of P-PDS owner, as a P-PDS owner may have more interest for some dimensions, say (s_t, o), than for other dimensions, say $(s_t, d), (s_t, DC), etc.$ Knowing about this "preference" would let the system adjust the final decision, giving more relevance to the dimension user cares more. In contrast, in such a situation, traditional ensemble might result in false positives/false negatives, as it is not able to catch user preferences in case of conflicting scenarios.

To overcome this problem, we propose an alternative strategy for aggregating the class labels returned by classifiers $\Theta_{ensemble}$. According to this approach, we assign a personalized weight to each single classifier in $\Theta_{ensemble}$, to reflect its relevance in the user opinion. As shown in Figure 6.2, we call this approach *personalized semi-supervised ensemble learning approach* (see Section 5.4 for more details).

Adopting this solution implies that when a new access request $\overline{AR}$ arrives, the P-PDS first collects the class values returned by $\Theta_{ensemble}$. If these are not conflicting, the P-PDS exploits the traditional ensemble approach for computing the final decision, otherwise it exploits personalized weights, as shown in Figure 6.2.

5.3 History-based Active Learning (HBAL)

As mentioned in Section 5.2, we propose to use AL but suggesting to increase the uncertainty value of a newly arrived access request $\overline{AR}$, based on its data consumer and service type. The key idea is that more the data consumer/service type in $\overline{AR}$ is different from data consumer/service type contained in access requests already labeled by the P-PDS owner, more its uncertainty level should be increased.

In doing so, the first step is the definition of two metrics able to quantify the distance between two service types/data consumers. In the following, we first introduce the distance metric for service type, then we discuss the metric for data consumer. Finally, we describe how we combine them to tune the AL uncertainty.

Service type distance. For this metric, we exploit semantic tools. In particular, we exploit the OWL-S framework [65] that mainly uses three types of sub-ontologies for semantically describing web services [64]. The first is the *service profile* sub-ontology, which describes the service provided to clients by a web service. The second is the *service process* sub-ontology to describe service functionalities and to specify how a client should interact with the service to get its functionalities. Finally, the *service grounding* sub-ontology describes how to invoke the web service. Among these three, the more relevant is the *service profile* sub-ontology, which contains properties such as *serviceName* (i.e., name of the service), *textDescription* (i.e., a brief description of the service). This latter includes also the description of the data the service requires as input and any other information the provider wants to include to describe the web service. In addition, the service profile sub-ontology uses *hasparameter* property to identify whether a service is being inherited by other services. By considering these relationships, it is possible to create an hierarchy among service types. Utilizing this hierarchy, we have been able to build the *service type hierarchy*, depicted in Figure 5.2.

Thus, exploiting this hierarchy we can measure the distance between two service types st_1 and st_2, denoted as $d_{ST}()$, as the number of edges in the path connecting the two leaves corresponding to st_1 and st_2.

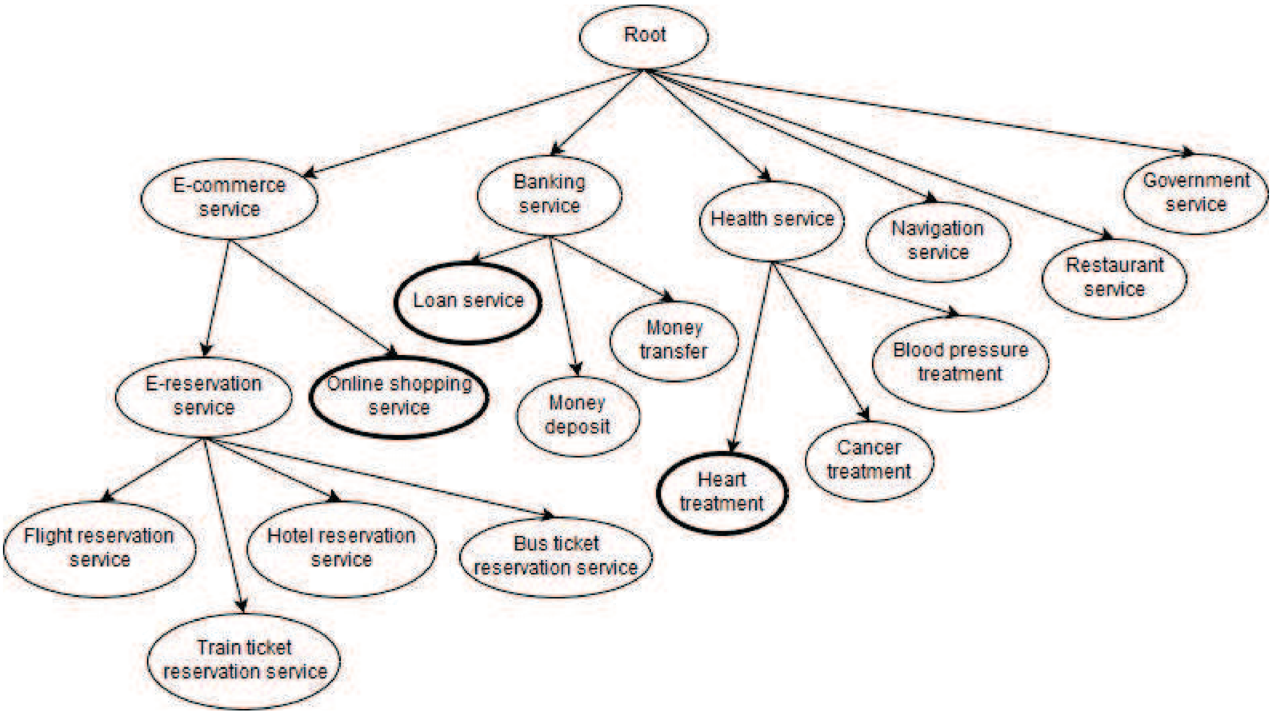


Figure 5.2: Service-type ontology

Data Consumer distance. For measuring the distance between data consumers, we assume that the system associates with each data consumer a profile consisting of a set of attributes describing the entity organization. For simplicity, in this chapter we assume this profile consists of the following attributes: *business category (BC)*, describing the type of the business the organization has; *origin (O)*, stating the country where the entity is based; and the *founded year (FY)*. However, different profile attributes can be considered as well.

Based on the profile, we measure the distance, denoted as $d_{DC}()$, between two data consumers DC_1 and DC_2 as the average of the distances between each attribute in their profiles. At this purpose, we exploit different distance metrics, depending on the nature of the attributes. For instance, for numeric values, the distance is calculated as the normalized Euclidean distance. For categorical attributes, we exploit the Jaccard similarity [48], if attribute values are not hierarchical organized. In contrast, if e attribute values are hierarchical organized, we can exploit existing distance measures defined for hierarchies (e.g., [68]).[1]

Uncertainty tuning. Given a new access request $\overline{AR}$ the idea is to adjust the uncertainty value associated with $\overline{AR}$, based on values of its data consumer and service type, more precisely based on their distances with respect to access requests already labeled by

[1] Other distance metrics can be used as well.

Algorithm 9 History-based_AL

```
**Input:** $LAR$, the set of labeled access requests;
$UAR$, the set of unlabeled access requests;
$\overline{AR}$, the newly arrived access request;
**Output:** $\Theta_{ensemble_{HBAL}}$
Let $SN$ be the number of labeled $AR$s required for building the preliminary model
if ($|LAR| < SN$) then
    Ask P-PDS owner a label for $\overline{AR}$
    Insert $\overline{AR}$ and its label into $LAR$
else
    Let $Rel$ be the set of pairs (m, n), where m and n are $\overline{AR}$ fields
    Let $\tau$ is the marginal threshold value among classes
    $UAR' = (UAR \cup \{\overline{AR}\})$
    $\Theta_{ensemble_{HBAL}}$=$Ensemble(LAR, UAR')$
    $Z_{w_{yes}} = \frac{1}{|Rel|} \sum_{Z=1}^{|Rel|} P_Z(Z_{yes}|\overline{AR})$
    $Z_{w_{no}} = \frac{1}{|Rel|} \sum_{Z=1}^{|Rel|} P_Z(Z_{no}|\overline{AR})$
    $Zw_{maybe} = \frac{1}{|Rel|} \sum_{Z=1}^{|Rel|} P_Z(Z_{maybe}|\overline{AR})$
    Let $nearest.st$ be the service type with the shortest distance to $\overline{AR}.st$
    Let $LAR_{nearest.st} \subseteq LAR$ be the set of access requests having $nearest.st$ as service type
    Let $DC_{values}$ be the set of data consumers specified in access requests in $LAR_{nearest.st}$

    Let $MostSimilar.DC$ be the data consumer in $DC_{values}$ having maximum similarity with $\overline{AR}.DC$
    $P = \frac{d_{DC}(\overline{AR}.DC, MostSimilar.DC)}{1 + d_{ST}(\overline{AR}.st, nearest.st)}$
    $\hat{Z}_{w_{yes}} = P * Z_{w_{yes}}$
    $\hat{Z}_{w_{no}} = P * Z_{w_{no}}$
    $\hat{Z}_{w_{maybe}} = P * Z_{w_{maybe}}$
    Let $Rank$ be a vector initialized with { $\hat{Z}_{w_{yes}}$,$\hat{Z}_{w_{no}}$,$\hat{Z}_{w_{maybe}}$} values
    Sort $Rank$ elements in descending order
    if ($|Rank[1] - Rank[2]| \leqslant \tau$) then
        Request P-PDS owner to provide a label for $\overline{AR}$
        Insert received label and $\overline{AR}$ into $LAR$
    else
        insert $\overline{AR}$ into $UAR$
        Evaluate_access_ request ($UAR$, $\Theta_{ensemble_{HBAL}}$)
    end if
end if
```

the P-PDS owner (i.e., the training dataset, denoted in what follows as LAR).

However, computing the distance between $\overline{AR}$ and each access request in the training dataset might be too time consuming. To avoid this waste of computation, we consider only the labeled access requests that offer the same (or similar) service type of the newly arrived request $\overline{AR}$. The underlying idea is that if two requests offer similar services also the corresponding data consumers should present some similarities in their profiles (aka in the business category field). If not, this might represent an anomaly that would be better to be evaluated directly by the P-PDS owner, i.e., its uncertainty value should be tuned.

Thus, a first step is to retrieve those labeled access requests in LAR offering a service type similar to the one in $\overline{AR}$. In support of this selection, the system traces in the service-type ontology tree those service types previously labeled by the P-PDS owner. More precisely, the system marks a leaf in the tree if there exists at least an access request in the training dataset that contains the value associated with that leaf node. Then, when a new access request $\overline{AR}$ arrives, our model checks whether the leaf corresponding to its service type is marked. If so, the distance $d_{ST}()$ is set to zero. Otherwise, the system measures the distance among this unmarked leaf and all other marked leaves, representing service types for which the P-PDS owner has expressed a label. Finally, among these distances, our model selects those leaves with the shortest distance. We denote the selected service type as $nearest.st$.[2]

Example 5.3.1. Let us consider, for instance, the service-type ontology described in Figure 5.2, assuming that only leaves $\{Loan, Online\ shopping, Heart\ treatment\}$ have been marked. Furthermore, let assume that $Flight\ reservation$ is the service-type associated with $\overline{AR}$. The system computes the following distances: $d_{ST}(Flight\ reservation, Loan)$=5, $d_{ST}(Flight\ reservation, Online\ shopping)$=3, and $d_{ST}(Flight\ reservation, Heart\ treatment)$=5. Among these distances, our model selects $d_{ST}(Flight\ reservation, Online\ shopping)$.

Exploiting the selected service types, the system retrieves the subset of access requests in the training dataset, denoted as $LAR_{nearest.st}$, having as value of service type the one with shortest distance with $\overline{AR}$, aka *nearest.st*. Following the previous example, $LAR_{nearest.st}$ consists of all labeled access requests with the *Online shopping* service type. Then, the system measures the distance between data consumer specified in $\overline{AR}$ and each data consumer specified in the access requests in $LAR_{nearest.st}$. Among these distances, the system selects the data consumer with the maximum similarity with $\overline{AR}$.DC. Hereafter, the selected data consumer is denoted as $MostSimilar.DC$.

Finally, to adjust $\overline{AR}$'s uncertainty value we exploit both the distance with service type *nearest.st* (i.e., $d_{ST}(\overline{AR}.st, nearest.st)$), and the distance with the data consumer $MostSimilar.DC$ (i.e., $d_{DC}(\overline{AR}.DC, MostSimilar.DC)$). In doing this, we exploit the following formula to compute the penalization value P:

$$P = \frac{d_{DC}(\overline{AR}.DC, MostSimilar.DC)}{1 + d_{ST}(\overline{AR}.st, nearest.st)}$$

[2] For simplicity, in what follows we assume that there exists only one service type with the shortest distance. Managing multiple service types with the shortest distance is an easy extension.

where $0 =< P <= 1$.

More precisely, this formula ensures that when $\overline{AR}$ contains a service type and a data consumer whose values have been both already labeled by the P-PDS owner (i.e., $d_{ST}()$=0 and $d_{DC}()$=1), then the tuning value P is set to 1. As it will be explained after, this implies that no penalization to the uncertainty value is applied, whereas $P < 1$ implies a penalization of the uncertainty value. The increase of the distance (i.e., increase of $d_{ST}()$ and decrease of $d_{DC}()$) implies smaller value of P (i.e., a greater penalization).

The pseudo code implementing HBAL is presented in Algorithm 9. The algorithm takes as input LAR (i.e., set of labeled access requests), UAR (i.e., the set of unlabeled access requests), $\overline{AR}$ (i.e., the newly arrived access request), and returns as output the updated $\Theta_{ensemble_{HBAL}}$ parameters values for each classifier.

According to the active learning approach, we first need to build a preliminary model based on a small number of labeled instances, denoted as SN. As such, until the LAR cardinality is less than SN, for every new access request Algorithm 9 simply asks the P-PDS owner to label it and insert the obtained label into LAR (see lines 3-4 in Algorithm 9). Once LAR is bigger enough, the arrival of a new access request will prompt the Algorithm to compute the $\Theta_{ensemble_{HBAL}}$ parameters (line 9). For doing so, for each dimension (i.e., $Rel =$ (d_0,DC), (d_0,p),$(d_0,\ s_t)$, $(d_0,\ o)$, (p,DC),(p,s_t),(p,o), (s_t,DC), (s_t,o),$(o,DC)\}$), Algorithm 9 computes the classifiers' parameters, $\Theta_{ensemble_{HBAL}}$, exploiting the $Ensemble()$ function (see Algorithm 7). Once the classifiers have been built, Algorithm 9 computes, for each class (i.e., yes, no, maybe), the sum of probabilities that $\overline{AR}$ has this label according to the dimensions in Rel (see lines 10-12).

The class probabilities computed in lines 10-12 are then penalized based on P value. To compute this value, Algorithm 9 has to first retrieve the service type that is the most similar to the one in $\overline{AR}$ (line 13), and, among the access requests in $LAR_{nearest.st}$, it retrieves the data consumer which is most similar to the one in $\overline{AR}$ (lines 16). Once P has been computed, $\overline{AR}$ probabilities are adjusted (lines 18- 20). Then, Algorithm 9 initializes a vector $Rank$ with values of these probabilities in descending order (lines 21-22). If the two classes with smaller probabilities have a distance smaller than τ[3] the system considers $\overline{AR}$ as uncertain and requests the P-PDS owner to insert a new label for $\overline{AR}$. Otherwise, $\overline{AR}$ is inserted into the set of unlabeled access requests UAR.

5.4 Personalized History-based Active Learning (PHBAL)

As discussed in Section 5.2, we propose to change the way ensemble aggregates labels returned by $\Theta_{ensemble_{HBAL}}$ classifiers in case of conflict. More precisely, we say that an access request has a *conflicting decision* if the returned labels are not within the same class (*yes*, *no*, or *maybe*). We recall that traditional ensemble approach does not make a distinction for conflicting classes, by computing the final decision simply aggregating

[3] In this setting, we assume the fixed uncertainty strategy [121]. It corresponds to label the instances for which the certainty is below some fixed threshold τ.

probabilities returned by classifiers. However, this does not take into account the semantics associated with each decision (aka class label).

Indeed, a user might give more relevance to some class labels, and less to others (e.g., given more relevance to no than yes labels). These preferences should be considered in resolving the conflicts. To take into account this, given $\overline{AR}$ we propose to adjust the probabilities returned by the classifiers based on a set of weights measuring the relevance the P-PDS owner gives to class labels. Here, the challenge is to properly set these weights. At this purpose, we propose to learn them analyzing the training set LAR, that is, the set of access requests directly labeled by P-PDS owners. In particular, we consider only those access requests in LAR, denoted as LAR_{conf}, that would be considered conflicting if analyzed by the classifiers (i.e., those access requests having labels not in the same class). More formally, given an $AR' \in LAR_{conf}$, we denote with $OL_{AR'}$ the owner label specified for AR', and with $CL_{AR'}$ the final computed label. The key idea is to find the set of weights that if applied to probabilities returned by classifiers would result in a final label $CL_{AR'}$ that maximizes the accuracy, that is, with the smallest difference w.r.t $OL_{AR'}$.

In order to learn more meaningful weights, we do not limit ourselves to simply consider the three class labels returned by the classifiers (i.e., yes, no, maybe) as discussed in Chapter 4. Indeed, since we are interested in measuring their relevance, we wish to take into account also their *strength*. For this purpose, we exploit the value of covariance in the probability distribution to categorize the class label of each classifier as *strong*, *moderate*, and *weak*. More precisely, when a classifier assigns to an access request a class label whose probability appears in the first deviation of the probability distribution, we consider the class label for that classifier as strong (e.g., $strong_{yes}, strong_{no}, strong_{maybe}$). Similarly, when a classifier assigns a class label for an access request whose probability appears in the second and the third deviation of probability distribution, we consider this class label as moderate (e.g., $moderate_{yes}, moderate_{no}, moderate_{maybe}$) and weak (e.g., $weak_{yes}, weak_{no}, weak_{maybe}$), respectively.

As an example, Figure 5.3 depicts the probability distribution of the *yes* class label predicted by the classifier $\Theta_{(o,p)}$ (i.e., for offer (o) vs purpose (p) dimension) over a training dataset. X-axis represents the offer values, whereas Y-axis represents the purposes converted into numerical values. More precisely, in Figure 5.3, the first, second and third circles enclose access requests of the training dataset, whereas classifier $\Theta_{(o,p)}$ returns the probabilities for *yes* class label with values in the first, second, and third deviations, respectively (i.e., $strong_{yes}$, $moderate_{yes}$, $weak_{yes}$).

Considering the strength allows us to have 9 different class labels, hereafter called as *class strength*. Thus, we find the set of weights to be applied to the classifiers, based on their class strength. Note that, if classifier $C1 = (d_0, o)$ and classifier $C2 = (s_t, o)$ return the same class strength they will have the same weight. More precisely, to find these weights, for each $AR' \in LAR_{conf}$, we iteratively set a random value for each weight, compute the obtained decision (i.e., $CL_{AR'}$) and check it against user's given label (i.e., $OL_{AR'}$). Among all iterated random values, we select the ones that maximize the accuracy. Algorithm 10 illustrates how the P-PDS determines the weight for each class strength returned by classifiers. It starts by computing the set W containing all possible combinations of values

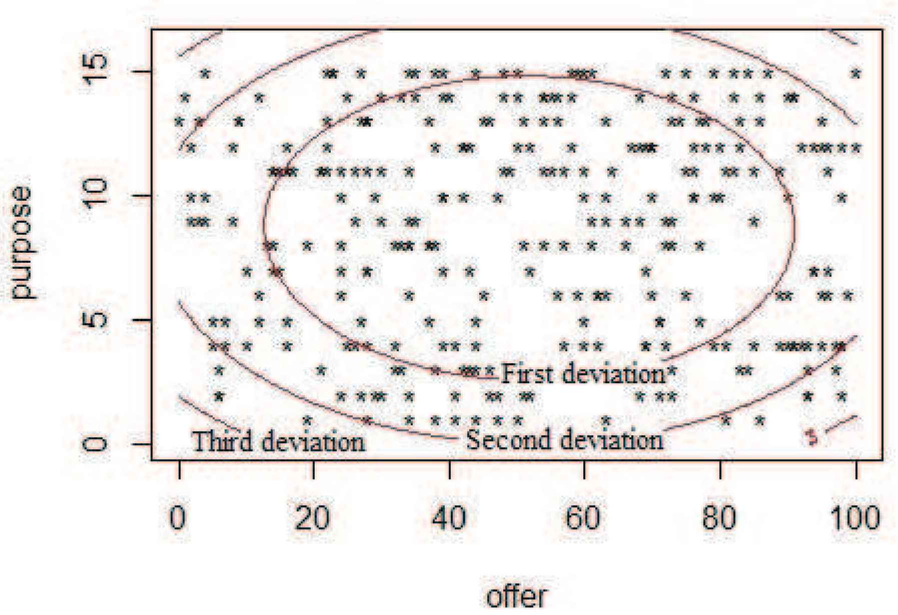


Figure 5.3: Class labels's classification based on deviation

for the 9 weights, where each value is selected in the set {1,2,3,4,5} (see line 1). Then, for each combination $w \in W$ and for each access request $AR' \in LAR_{conf}$ Algorithm 10 does the following: it generates the 9 class strengths produced by classifiers $\Theta_{ensemble_{HBAL}}$ (line 4); it counts the occurrences of each class strength (lines 10- 27); it computes the weighted sum of probabilities of the three main classes (yes, no, maybe), by applying the iterated combination of weights w (lines 30- 32); and based on these personalized aggregated probabilities Algorithm 10 takes a decision $CL_{AR'}$ for AR' (lines 33-39).

This decision is then compared to the one assigned by the P-PDS owner, that is $OL_{AR'}$. If the accuracy obtained using the computed weights w is higher or equal to the ones achieved by previous weights, Algorithm 10 stores w into $weight$ and proceeds to elaborate another conflicting access request in LAR_{conf} (lines 40-45). Once all possible weights and access requests have been evaluated, Algorithm 10 processes the stored values in $weight$ by computing the average value for each component (see lines 48-49). These values are then returned to Algorithm 11 to estimate class label of upcoming access requests.

Algorithm 10 is exploited by the P-PDS to initialize the personalized weights. This is done when the P-PDS has to evaluate a new access request. Indeed, if the newly arrived access request $\overline{AR}$ has not to be labeled by P-PDS owner, it is inserted into the UAR set which is then evaluated by P-PDS using the updated classifiers (cfr. Algorithm 11). This evaluation is done by Algorithm 11, which exploits the personalized semi-supervised ensemble approach for those access requests in UAR having a conflicting decision (see lines 23-37); otherwise, it exploits traditional ensemble.

More precisely, the final prediction is then computed by combining each class probability

$P_{\overline{AR}}$, predicted by each single classifier $\Theta_{ensemble_{HBAL}}$ based on bagging method [37, 79].[4] The final class label is set as the one with maximum probability (see lines 39 - 42 in Algorithm 11).

[4] Bagging is an effective method for ensemble learning, where the final label is assigned by computing the average of membership probabilities returned by the obtained classifiers.

Algorithm 10 Personalized_weight ()

Input: $LAR_{conf} \subseteq LAR$, **the set of conflicting access requests in the training dataset**

Output: $weight$, **the vector storing the weights for each class strength**

```
Let W be the set of arrays in the Cartesian product $\times_{k=1}^{9} D_k$, where $D = \{1,2,3,4,5\}$
for each $w \in W$ do
  for each $AR' \in LAR_{conf}$ do
    Let $C_s$ be the set of label strengths returned by classifiers $\Theta_{ensemble_{HBAL}}$ on $AR'$
    Let $OL_{AR'}$ be the class label given by the PDS owner to $AR'$
    Let CS be a vector of 9 elements initialized to 0
    Let $Acc$, $sum_{yes}, sum_{no}, sum_{maybe}$ be initialized to 0
    for each $x \in C_s$ do
      switch $(x)$
      case "$strong_{yes}$":
        CS[1]=CS[1]+1
      case "$moderate_{yes}$":
        CS[2]=CS[2]+1
      case "$weak_{yes}$":
        CS[3]=CS[3]+1
      case "$strong_{no}$":
        CS[4]=CS[4]+1
      case "$moderate_{no}$":
        CS[5]=CS[5]+1
      case "$weak_{no}$":
        CS[6]=CS[6]+1
      case "$strong_{maybe}$":
        CS[7]=CS[7]+1
      case "$moderate_{maybe}$":
        CS[8]=CS[8]+1
      default:
        CS[9]=CS[9]+1
      end switch
    end for
    $sum_{yes} = sum_{yes} + CS[1] * w[1] + CS[2] * w[2] + CS[3] * w[3]$
    $sum_{no} = sum_{no} + CS[4] * w[4] + CS[5] * w[5] + CS[6] * w[6]$
    $sum_{maybe} = sum_{maybe} + CS[7] * w[7] + CS[8] * w[8] + CS[9] * w[9]$
    if $(sum_{yes} \geqslant sum_{no}) \wedge (sum_{yes} \geqslant sum_{maybe})$ then
      $CL_{AR'} = yes$
    else if $(sum_{no} \geqslant sum_{yes}) \wedge (sum_{no} \geqslant sum_{maybe})$ then
      $CL_{AR'} = no$
    else
      $CL_{AR'} = maybe$
    end if
    $acc = accuracy(CL_{AR'}, OL_{AR'})$
    if $acc \geqslant Acc$ then
      $weight = w$
      $Acc = acc$
    end if
  end for
end for
Let $AvgWeight$ be a vector of 9 elements initialized as empty
for $i \in \{1, 2, ...9\}$ do
  $AvgWeight[i] = \frac{1}{|weight|} \times \sum_{\forall \overline{w} \in weight} \overline{w}[i]$
end for
Return: $AvgWeight$
```

Algorithm 11 Evaluate_access_request()

```
Input: $UAR$, the set of unlabeled access request;
Output: Labels for $UAR$
Let $C_S$ be a vector of $|\Theta_{ensemble_{HBAL}}|$ elements
Let $P_{\overline{AR}}(yes), P_{\overline{AR}}(no), P_{\overline{AR}}(maybe)$ be initialized to 0
Let $i$ be initialized to zero
for each $uar \in UAR$ do
  for $\theta \in \Theta_{ensemble_{HBAL}}$ do
    Let $P^{\theta}_{\overline{AR}}(yes)$ be the probability for $\overline{AR}$ of being labeled as yes, computed using $\theta$
    Let $P^{\theta}_{\overline{AR}}(no)$ be the probability for $\overline{AR}$ of being labeled as no, computed using $\theta$
    Let $P^{\theta}_{\overline{AR}}(maybe)$ be the probability for $\overline{AR}$ of being labeled as maybe, computed using $\theta$
    if $(P^{\theta}_{\overline{AR}}(yes) \geqslant P^{\theta}_{\overline{AR}}(no)) \wedge (P^{\theta}_{\overline{AR}}(yes) \geqslant P^{\theta}_{\overline{AR}}(maybe)$ ) then
      $C_S[i] = yes$
    else
      if $(P^{\theta}_{\overline{AR}}(no) \geqslant P^{\theta}_{\overline{AR}}(yes)) \wedge (P^{\theta}_{\overline{AR}}(no) \geqslant P^{\theta}_{\overline{AR}}(maybe))$ then
        $C_S[i] = no$
      else
        $C_S[i] = maybe$
      end if
    end if
    $i = i + 1$
    $P_{\overline{AR}}(Yes) = P_{\overline{AR}}(Yes) + P^{\theta}_{\overline{AR}}(Yes)$
    $P_{\overline{AR}}(No) = P_{\overline{AR}}(No) + P^{\theta}_{\overline{AR}}(No)$
    $P_{\overline{AR}}(Maybe)$= $P_{\overline{AR}}(Maybe) + P^{\theta}_{\overline{AR}}(Maybe)$
  end for
  if (decisions are conflicting in $C_S$) then
    Let CS be a vector of 9 elements represents the class strengths on $\overline{AR}$
    Let $w$ be a vector of 9 elements
    $w$ = Personalized_weight_algorithm()
    $sum_{yes} = CS[1] * w[1] + CS[2] * w[2] + CS[3] * w[3]$
    $sum_{no} = CS[4] * w[4] + CS[5] * w[5] + CS[6] * w[6]$
    $sum_{maybe} = CS[7] * w[7] + CS[8] * w[8] + CS[9] * w[9]$
    if $(sum_{yes} \geqslant sum_{no}) \wedge (sum_{yes} \geqslant sum_{maybe})$ then
      $CL_{ar'} = yes$
    else if $(sum_{no} \geqslant sum_{yes}) \wedge (sum_{no} \geqslant sum_{maybe})$ then
      $CL_{ar'} = no$
    else
      $CL_{ar'} = maybe$
    end if
    Return:$CL_{ar'}$
  else
    $P_{\overline{AR}}(yes)$= $P_{\overline{AR}}(Yes)/|\Theta_{ensemble_{HBAL}}|$
    $P_{\overline{AR}}(no)$= $P_{\overline{AR}}(No)/|\Theta_{ensemble_{HBAL}}|$
    $P_{\overline{AR}}(maybe)$= $P_{\overline{AR}}(Maybe)/|\Theta_{ensemble_{HBAL}}|$
    Return: $C_D$, and the label corresponding to max($P_{\overline{AR}}(yes)$,$P_{\overline{AR}}(no)$,$P_{\overline{AR}}(maybe)$)
  end if
end for
```

5.5 Experiments

To demonstrate the effectiveness of the proposed P-PDS, we conduct several experiments. In the first experiment, we measure the accuracy level and F1 score obtained using the semi-supervised ensemble (SSE) [85], semi-supervised active learning (SSAL), history-based active learning (HBAL), and personalized history-based active learning (PHBAL).

In the next experiment, we analyse the access requests which are selected by SSE, SSAL and HBAL to be labeled directly by the P-PDS owners. With this experiment, we want to see which learning approach needs more training data to learn users' privacy preferences.

Finally, we compute the satisfaction level of P-PDS owners regarding the decisions taken by the various learning approaches. Moreover, we consider user quality in terms of feedback on the training dataset to investigate how a badly labeled training dataset impacts user satisfaction.

5.5.1 Settings

In this Chapter, we have created a dataset consisting of 303 access requests, by using realistic values for data consumer, service type, requested data field, purpose and offer value as did in Section 4.4.1. But here we have generated another dataset considering more elements compared to 4.4.1. More precisely, we have considered 55 different data consumer profiles; 18 different service types; 42 possible data fields; 21 purposes; and offer values ranging from 0% to 100%. Based on these elements we randomly generate access requests.

To collect labels for the training dataset, as well as P-PDS owner feedback on access request decisions, we designed a web application. To ensure the involvement of a good number of participants, acting as P-PDS owners, we exploited a crowd-sourcing platform. More precisely, we used the Microworker crowd-sourcing platform[5] for the enrollment of participants (called workers) of different nationalities, ages, and educational levels. At this purpose, we selected only the workers with the best rating according to the Microworker platform. Once the job has been accepted, each worker has been redirected to our web application to conduct both the learning phase (aka, labeling the training dataset) and the testing phase (aka evaluating the P-PDS decisions). As further quality check, we measured the time each participant devoted to the learning task and, if this is less then a reasonable time, we removed the participant. We obtained data from 360 workers.

In order to measure the effectiveness of the proposed learning approaches, we use the same traditional confusion matrix given in Table 4.2 to define the evaluation metric as shown in Table 4.3 to measure the effectiveness of the proposed approaches.

5.5.2 Results

Effectiveness

In this experiment, we show a comparative analysis of accuracy obtained by SSE, SSAL, HBAL and PHBAL.

[5] https://www.microworkers.com

Table 5.1: Comparison of SSE, SSAL, HBAL, and PHBAL approach for the training dataset

	SSE			SSAL			HBAL			PHBAL		
	Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe
Precision	86.71 %	71.03%	51.74%	91.65%	66.64%	50.97%	91.10%	67.35%	52.06%	94.10%	71.22%	60.79%
Recall	79.84 %	74.51%	63.83%	83.45%	77.82%	65.83%	82.10%	77.26%	69.67%	84.73%	81.93%	81.35%
F1	83.13 %	72.73%	57.16%	87.36%	71.80%	57.45%	86.37%	71.97%	59.59%	89.17%	76.20%	69.58%

Table 5.2: Comparison of SSE, SSAL, HBAL, and PHBAL approach for the testing dataset

	SSE			SSAL			HBAL			PHBAL		
	Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe	Yes	No	Maybe
Precision	85.24%	74.09%	64.28%	89.48%	75.58%	58.88%	88.26%	74.86%	68.68%	90.43%	80.79%	58.58%
Recall	86.86%	76.89%	55.38%	90.29%	69.89%	64.29%	89.69%	77.01%	60.17%	90.75%	86.67%	51.32%
F1	86.04%	75.46%	59.50%	89.88%	72.63%	61.46%	88.97%	75.92%	64.15%	90.59%	83.63%	54.17%

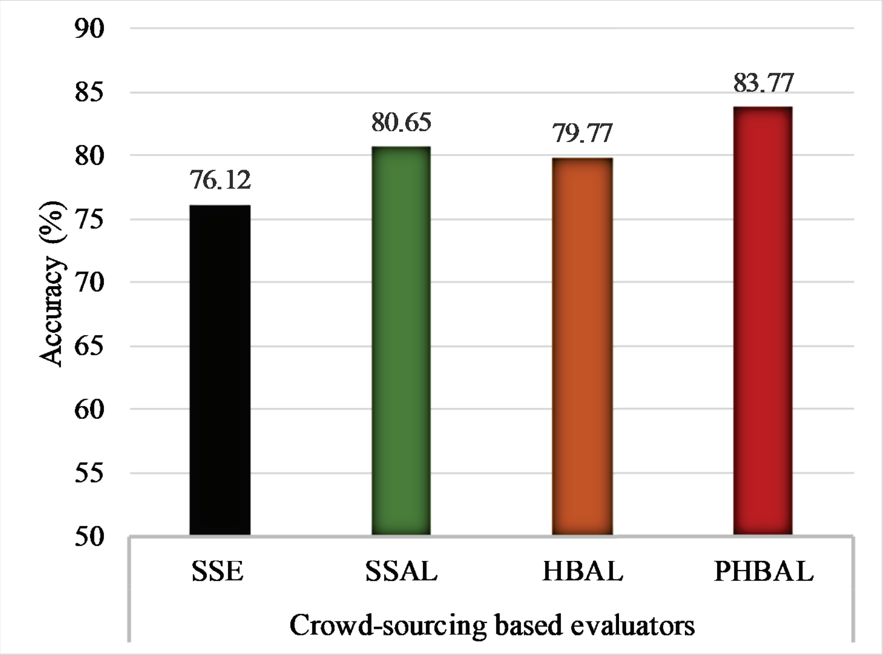


Figure 5.4: Accuracy of SSE, SSAL, HBAL, PHBAL

Accuracy. As a first experiment, we have run each learning approach on the training dataset LAR generated by workers, to compute the true positive values (i.e., TP_{yes}, TP_{no}, TP_{maybe}) and corresponding accuracy value (see Table 4.3). More precisely, we split the 360 participants into three groups: 120 participants have tested P-PDS with SSE learning approach; 120 with SSAL; and 120 with HBAL. As such, we obtain three training datasets, namely LAR_{SSE}, LAR_{SSAL}, and LAR_{HBAL}, generated by participants using P-PDS with SSE, SSAL, HBAL learning approach, respectively. Moreover, we have an additional training dataset, denoted as LAR_{PHBAL}, consisting of those access requests in LAR_{HBAL} that have been judged conflicting and thus have been treated according to the personalize history-based active learning approach (see Section 5.4). In the experiment, among the 120 access requests evaluated according the HBAL approach, 106 have been judged as conflicting by the classifier.

Figure 5.4 shows that obtained accuracy. We can see that around 76.12% of the training dataset is correctly labeled by SSE, around 80.65% by SSAL, 79.77% by HBAL, whereas 83.77% by PHBAL. We note that PHBAL and HBAL have a greater accuracy than SSE. More precisely, Figure 5.4 depicts that PHBAL can improve the accuracy of SSE (e.g., the proposal in [85]) of 7.65%.

However, we also notice that HBAL is little less accurate than SSAL. This is due to the fact that HBAL further penalizes the uncertainty value of access requests based on the values of service type and data consumer (see Section 5.3). Consequently, we expect that, based on distribution of values of service type/data consumer, P-PDS owner will

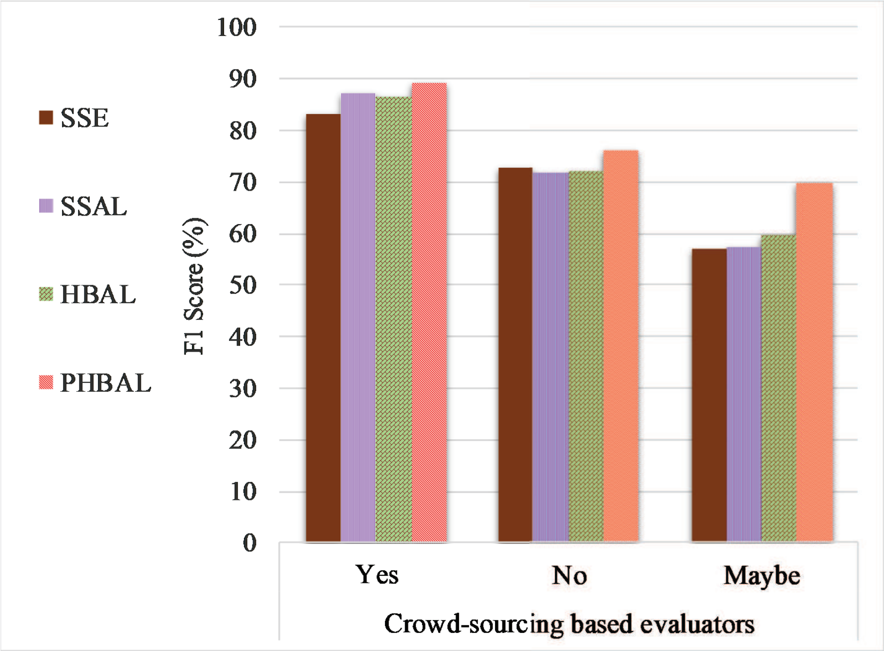


Figure 5.5: F1 score for training dataset

be asked to label a greater number of access requests compared to SSAL[6] and that this might have an impact on accuracy. To check how this impacts, we have looked on the false negatives/false positives produced by HBAL and SSAL, respectively. The results show that HBAL produces 3.47% false negatives and 7.14 % false positives, while SSAL has 3.27% and 7.16%, respectively.[7] This highlights that SSAL and HBAL have almost the same number of false positives, whereas HBAL errors are mainly increased due to false negatives. This implies that HBAL is more conservative w.r.t. the preservation of users, in that the majority of misclassification lead to the denial of authorized data release, rather than to the release of unauthorized data. PHBAL exploits the access requests included in *LAR* of HBAL as training dataset for predicting the class label of conflicting access requests. Figure 5.4 shows that PHBAL improves the accuracy of HBAL of 3.00%.

F1 score. We have also measured the F1 score for each class (yes, no, maybe) for comparing the performance among the learning approachers with respect to the training dataset and testing dataset, separately (see Tables 5.1 and 5.2). According to Table 4.3, F1 score can be measured by considering both precision and recall. The precision is the ratio of the number of correctly predicted items (i.e.,access requests) to the total number of predicted (correctly and incorrectly) items. Whereas, recall is the ratio of the number of correctly predicted items to the total number of relevant items. Our analysis shows in Figure 5.5 for training and Figure 5.6 for testing dataset respectively. Results show that

[6] We refer to next experiment, aka Figure 5.7 depicting the training dataset distributions.

[7] To compute false negatives/positives, we have considered only the class labels yes and no, since class label *maybe* represents users confusion to take decision on upcoming access requests.

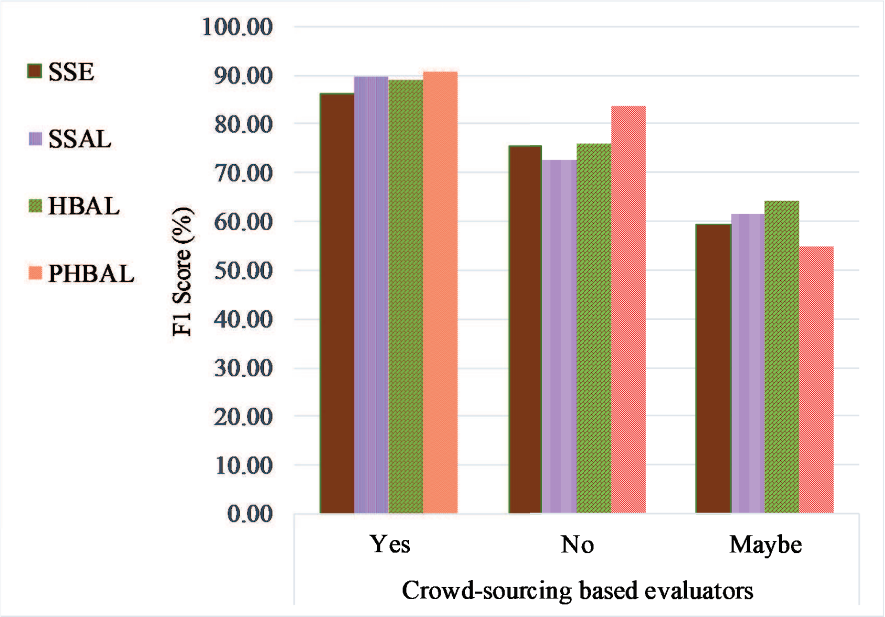


Figure 5.6: F1 score for testing dataset

PHBAL works comparatively better than other approaches. However, for testing dataset, it produces good accuracy in case of class *yes* and *no*, whereas for the class label *maybe*, its accuracy is lower than others approaches.

More specifically, PHBAL provides good precision and recall values for *yes* and *no* decisions in-spite of having poor precision for *maybe* decision for testing dataset. It is relevant to highlight that poor precision for *maybe* decision does not produce any threat to privacy preferences as P-PDS asks PDS owner for his/her final decision regarding on the *maybe* decisions.

Distribution of the training dataset

In this experiment, we observe the distribution of access requests selected by the P-PDS for being labeled by owner so as to be included in the training dataset *LAR*. We refer to this distribution as the training dataset distribution (TDD). The purpose is to compare TDD when P-PDS adopts the semi-supervised ensemble (SSE) [85], semi-supervised active learning (SSAL), and history-based active learning (HBAL) approaches. In this experiment, we do not consider personalized history-based active learning (PHBAL) approach because its training dataset is the same of HBAL. As described in previous experiment, we split the 360 participants into three groups: 120 participants have tested P-PDS with SSE learning approach; 120 with SSAL; and 120 with HBAL. For generating the TDD graph of different learning approaches, we analyze the stream of arriving access requests by grouping them in bunch of 5 access requests each (e.g., [1-5],[6-10],[11-15], etc.). Then, for each learning approach (i.e., SSE, SSAL, HBAL), as a first statistic, we have checked

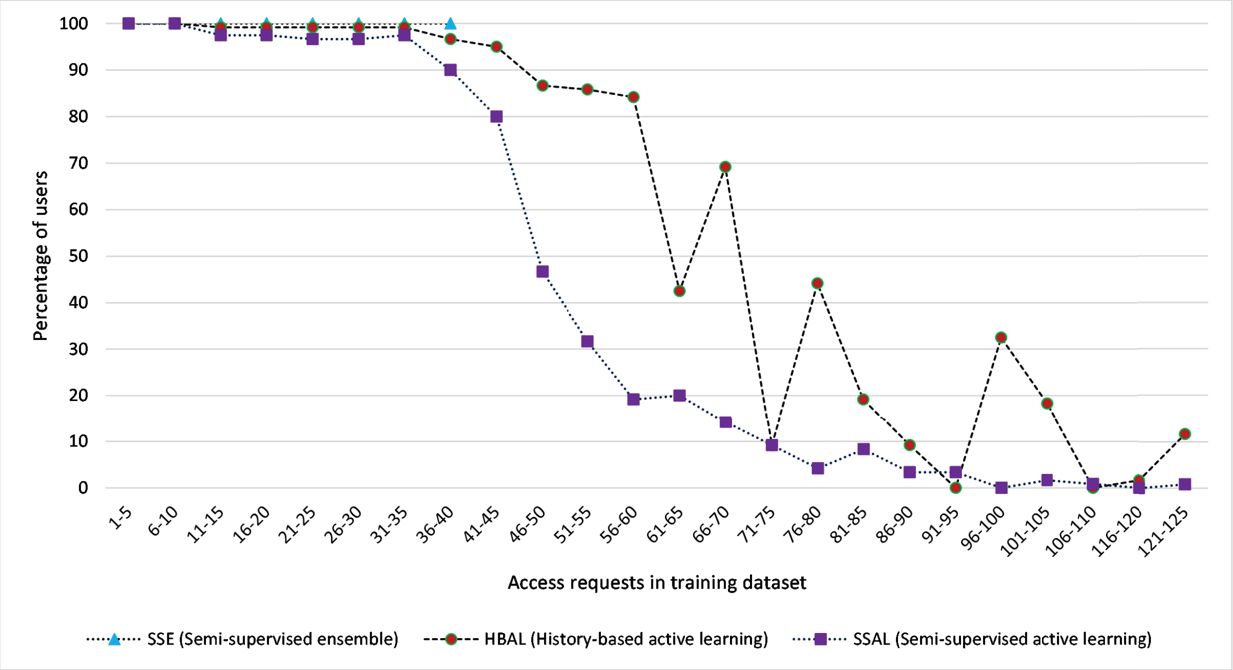


Figure 5.7: Training dataset distribution for SSE, HBAL, and SSAL

the number of users that have been asked to label at least an access request in each group. Figure 5.7 reports these results, where the X-axis shows the sequence of access requests' groups and the Y-axis the percentage of users that have been asked to label at least an access request (e.g., according SSAL approach, 20% of users have labeled at least a request in the access requests group [56-60]).

Regarding TDD of SSE method, we have to recall that this approach selects a fixed number of access requests as training dataset. In our experiments, we set this number, denoted as N_{SSE}, to 40, as shown in Figure 5.7. Regarding SSAL, this approach first builds a preliminary learning model based on a small number of access requests, which is typically smaller than N_{SSE}. Then, the obtained preliminary model is used to select the new access requests to be labeled by the P-PDS owner (i.e., to be inserted into LAR) based on their uncertainty values. In our experiments, we set this small number, denoted as N_{SAL}, to 9. As such, as reported in Figure 5.7, the P-PDS exploiting the SSAL approach selects the first 9 access requests for the training dataset. Then, from the 10th access request on, the P-PDS exploits the preliminary model and, as depicted in Figure 5.7, the percentage of owners that have to label at least an access requests decreases, and, as a consequence, the number of access requests the P-PDS is able to automatically label increases. As an example, from Figure 5.7, we can notice that in the frame [36-40], SSAL is able to make a decision for 10% of users, reaching the about 50% in the frame [46-50], whereas following the SSE approach users are forced to label all access requests in [1-40].

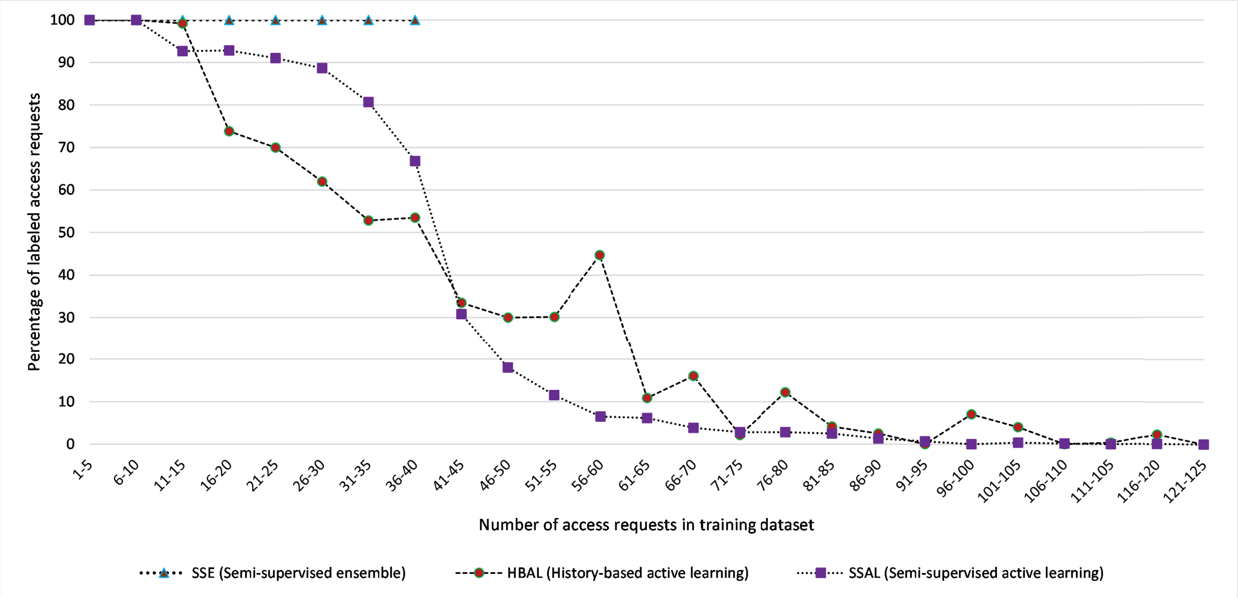


Figure 5.8: Training dataset distribution for SSE, HBAL, and SSAL

Regarding HBAL, we have to recall that this follows the AL approach, but modified so as to increase the uncertainty value based on the value of data consumer and service type of the newly arrived access requests (see Section 5.3).[8] As such, we expect that the TDD of HBAL: (1) shows the same benefit of the SSAL approach, in terms of reduced percentage of owners labeling access requests; (2) depends on the distribution of values of data consumer/service types, since uncertainty penalization does.

Figure 5.7 confirms these expectations. Indeed, as depicted in the figure, HBAL is able to automatically label access requests before the SSE but after the SSAL approach (e.g., HBAL is able to make a decision for about 10% of users in the frame [46-50], whereas SSE only after the 40th access request and SSAL in [36-40]). Moreover, TDD of HBAL shows different peaks (e.g., [66-70],[76-80], etc) as consequence of distribution of values of data consumer/service types (e.g., access requests in frame [66-70] contain data consumer/service types never labeled before). However, despite these peaks, Figure 5.7 shows that the number of access requests to be labeled by the P-PDS owner decreases.

In addition to the above-described statistics on the percentage of users that have labeled at least an access request in group, we have also analyzed the total number of access requests that each approach requires to be labeled by P-PDS owner. More precisely, considering that each learning approach has been tested by 120 participants, in each group we collect a total number of 600 access requests (e.g., 120× 5). Figure 5.8 shows the percentage of access requests that on the total of 600 access requests have been labeled by owners. This

[8] In the experiments, we build the preliminary learning model of HBAL using the same number of labeled access request we used for SSAL, i.e., $N_{HBAL} = 9$.

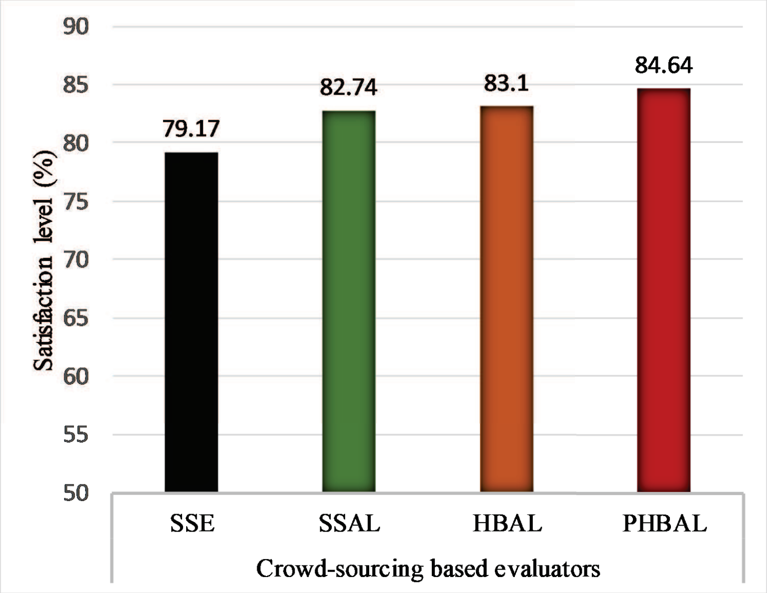


Figure 5.9: Evaluators satisfaction level

result further confirms the benefit of HBAL w.r.t. SSE, that is, the HBAL's capability to automatically label new access requests before reaching the fixed number of labeled access requests imposed by SSE, i.e., $N_{SSE} = 40$. Moreover, we can notice that even if HBAL implies a penalization on the uncertainty values, HBAL and SSAL presents similar trends, where the extra labels that, compared to SSAL, HBAL requires to owners brings benefits in the obtained effectiveness (see experiments in Section 5.5.2).

Participant Evaluation

In this experiment, we collect feedback from P-PDS owners regarding the decisions taken by the different learning approaches in order to evaluate their satisfaction.

Satisfaction level. For this purpose, we exploit the designed web application to show to each participant a predefined set of access requests with the decisions returned by the learning approaches, asking them to insert their own decisions. Then, to measure the participant satisfaction level we compute the true positive values (i.e., TP_{yes}, TP_{no}, TP_{maybe}). More precisely, the decision suggested by the learning approach is considered as a true positive if it is the same as the one inserted by the participant. Thus, we define satisfaction level as the ratio of the total number of true positives and the total number of evaluated access requests. More precisely, we have shown to each participant 10 access requests, where: 7 are new access requests, whose decisions have been generated by the learning approach; the remaining 3 are taken from the set of access requests labeled by that participant during the generation of the training dataset. These latter are used for checking the consistency of participant judgments, and thus making some considerations on the quality of their evaluation (see the discussion in the next section).

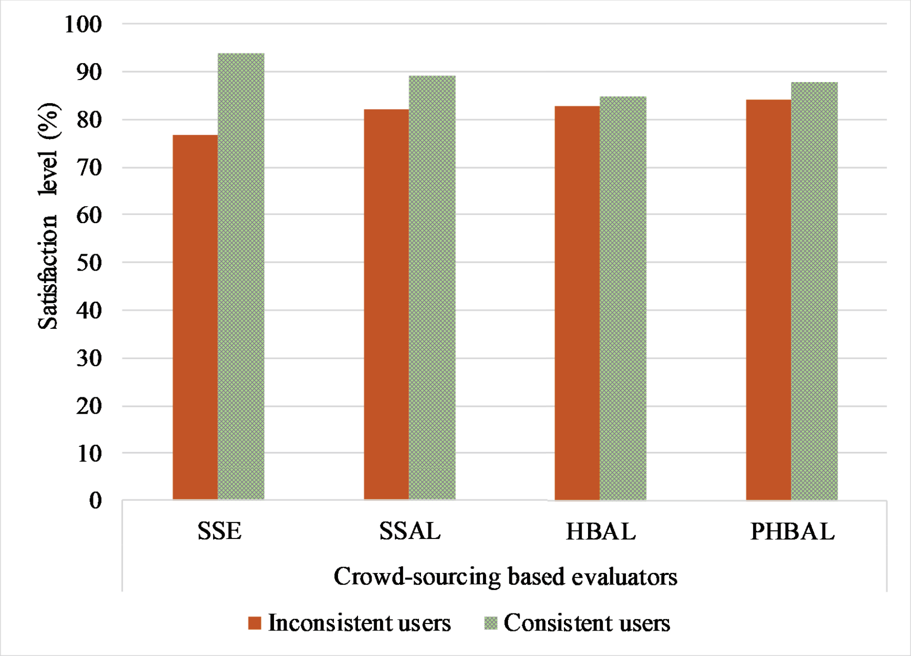


Figure 5.10: Satisfaction level of consistent and inconsistent evaluators

As shown in Figure 5.9, around 79.17% participants are satisfied with decisions taken by SSE, around 82.74% with SSAL, 83.10% with HBAL, and 84.64% are fine with the decisions taken using PHBAL. This experiment shows that participants are less satisfied by the performance of SSE than HBAL and PHBAL. Figure 5.7 reports the comparison.

Evaluator quality. The output of the machine learning approaches depend on P-PDS owner's input on the training dataset. Therefore, we are interested to investigate how a badly labeled training set impacts the final satisfaction level. Thus, we set some strategies to identify consistent and inconsistent evaluators. In this case, we follow two strategies: the first one is recalled from Section 4.4.4, where 3 access requests are taken from the set of access requests that evaluators have labeled during the first phase. In particular, in presenting these access requests during the testing phase, the web application shows, as taken decision, the same label entered by the evaluator during the first phase. Based on the satisfaction level the evaluator assigns to this label, we can judge whether the evaluator is consistent or not in his/her decisions, which gives us a measure of the quality of his/her jobs. Second, we have inserted 2 access requests in the first phase (e.g., among the first nine access requests) which contain inconsistent requested data fields with respect to the purpose and service. For example, we ask a label for an access request these data {*traveling date, traveling time, From (starting place), To (destination place), etc.,*}, having a service purpose *issuing a loan.* We expect that, in case of an inconsistent access request, a participant that carefully reads the request gives a deny decision/label. Based on these two strategies, we consider a participant as consistent if he/she behaves correctly w.r.t. the 3 access requests (i.e., (s)he is satisfied by the decisions shown by the web application, as they are the same (s)he has inserted during the generation of the training dataset) and (s)he denies the 2 inconsistent access requests.

Figure 5.10 presents the comparative analysis of the satisfaction level for consistent and inconsistent participants. The figure shows that the satisfaction level of consistent users is greater than the satisfaction level of inconsistent users. However, even in the worst case, about 70% of inconsistent evaluators are satisfied by the taken decisions.

5.6 Chapter summary

In this paper, we exploit active learning concepts to select the informativeness training dataset to improve the performance of learning classifiers. In-spite of having this motivation, we do not simple use the active learning approach in our context but also we propose a strategy to measure the uncertain access requests by considering elements (i.e., service type and data consumer) besides the strategy of traditional active learning approach, since we believe that for selecting most uncertain access requests as training dataset particularly rely on the elements namely service type and data consumer. With this direction, we impose more privacy restriction and improve performance of classifiers for predicting decisions on upcoming access requests in P-PDS. Furthermore, we also exploit the obtained classifiers decisions in term of measuring the strengths/weights by considering the accuracy of the classifiers with respect to training dataset. More precisely, we measure the weights vector for users privacy preferences based on the classifiers decisions which can be used for taking decision regarding upcoming newly arrival access requests whether it might be allowed to access P-PDS or not. With this way, we can minimize the false positive/false negative cases which certainly imply to improve the performance of learning classifiers.

Chapter 6

Contextual based privacy preference in PDS

6.1 Introduction

In the past few years, studies have shown that contextual information influences user's decision when sharing personal data with third parties [80,87,107]. Moreover, Nissenbaum's theory illustrates the reason for which most of the privacy preference models fail to protect the privacy preference violations is that they do not consider contextual integrity [77]. To date, several studies on PDS have suggested to enforce privacy preferences that regulate the third parties access to PDS [6,35,84,101] without considering user contextual information.

However, literature shows that users prefer to set his/her privacy preferences taking into account the contextual data [28, 73]. In fact, contextual information can be used to design privacy preference framework that can define privacy preference according to the user's current situation. Thus, it can improve the overall usability and level of control on personal data. Let us consider that a user may feel comfortable to take a decision, when (s)he is in travel, regard on an access request seeking current location of the user in term of suggesting some nearby famous places relevant to the user preferences for visiting. Thus, it is required to develop privacy preference mechanism that can leverage contextual data with non-contextual fields (e.g., access request elements) to learn user privacy preferences efficiently.

In this chapter, we describe a contextual based privacy preference mechanism for PDS. To figure out the latent correlations between the contextual data and user's opinion on the access request, we conduct experimental analysis on user privacy preference based on users' opinions on contextual based access request. With this intention, we exploit the context information of the user and access request elements to train up the learning model about user privacy preferences. Once the learning model has been built then it can predict the decision automatically on the newly arrived contextual based access request according to user contextual information. Moreover, in this paper, we also want to explore the mechanism to reduce the over-fitting problem occurs in machine learning approaches. In

Context={Day of week, Time of the day, Place, Activity}	
Day of week	{Workweek days, Weekend days}
Time of the day	{Morning (6.00-11.59), Afternoon(12.00- 17.59), Evening(18.00- 23.59), Night (0.00-5.59)}
Place	{Home, Office/School, Outside}
Activity/Feelings	{Meeting, Working, Running, Studying, Traveling, Eating, Sleeping, Idle, Physical Exercise, Driving, Sick}

Table 6.1: Contextual data

general, over-fitting occurs when a learning model learns the noise/randomness along with the samples in the training dataset that negatively impacts the performance of the learning model on the upcoming new samples. To reduce over-fitting, the general approach is to vary the number of training dataset sequentially and check the accuracy on the testing dataset. The fact is that those combination of training dataset produce better accuracy on testing dataset will be used for further predication. At this purpose, we proceed with a approach: first we select the total number of training dataset according to the history based active learning (see Chapter 5). After that we check which are the most uncertain instances (having probability difference of the class labels are very close) in the total number of training dataset and select the top 20 uncertain training dataset for a learning model and check the accuracy on the testing dataset. By the same way, we then consider top 25 uncertain training dataset and check the accuracy on testing dataset. Like this way, we proceed on and select the best model.

The rest of this chapter is organized as follows. Section 6.2 introduces the overall architecture of our proposed context privacy-aware PDS (CP-PDS) framework, whereas Section 6.3 illustrates the experimental results.

6.2 Context Privacy-aware PDS (CP-PDS)

The privacy preference framework presented in the Chapter 5 learns PDS owner's privacy preferences by exploiting only access request elements (e.g., data consumer, requested data, service type etc.). As a matter of fact, this approach could not fully cover all aspects of user concern in term of ensuring privacy on their personal data as it did not consider user contextual information. Indeed, user might change his/her mind w.r.t privacy management based on his/her present situation (e.g., contextual data) when the access request arrives to PDS. For example, let suppose that PDS owner U receives an access request offering the service *entertainment* during his/her office hours (e.g., office hours refer user's contextual information). In this case, U might always deny this access request. However, we believe this might bring to a bias, as, it may happen that the classifier is trained based on situations where users deny access requests that, in different contexts, might be accepted, and vice versa. As an example, let us consider an access request asking to obtain PDS data for some entertainment services. If during the training phase, these type of access requests come always during office hours, the PDS's owner most likely will deny, resulting in the

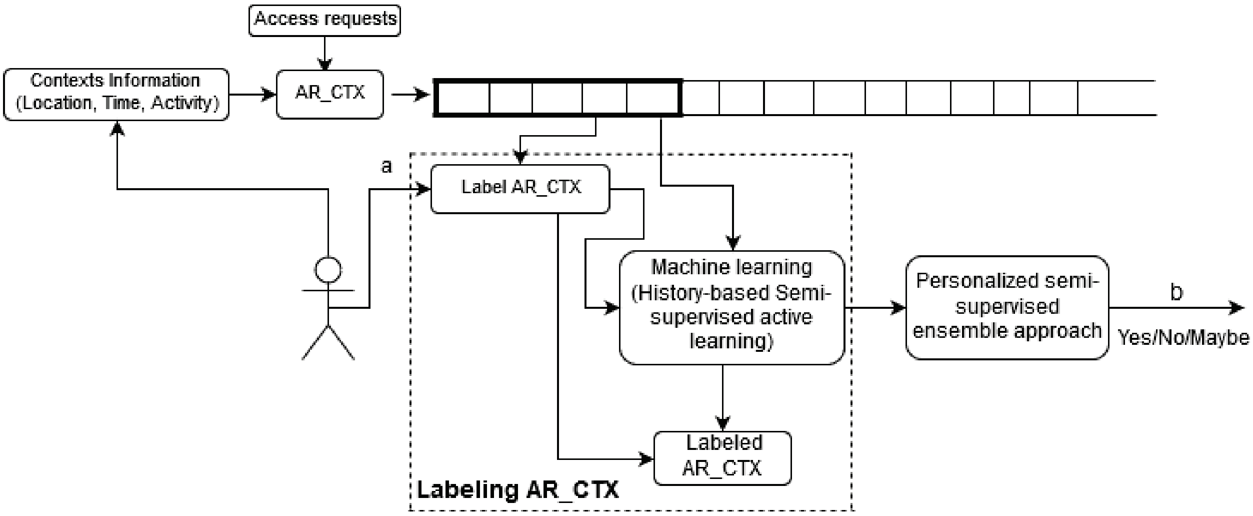

Access requests
Contexts Information (Location, Time, Activity)
AR_CTX
a
Label AR_CTX
Machine learning (History-based Semi-supervised active learning)
Personalized semi-supervised ensemble approach
b
Yes/No/Maybe
Labeled AR_CTX
Labeling AR_CTX

Figure 6.1: Context privacy-aware PDS (CP-PDS)

fact that the learning approach will train the classifier to always deny this type of services. However, PDS' owner might be willing to accept this type of services in other contexts, aka during his/her free time. Thus, if we do not consider contextual information to train up learning models then it will produce more prediction errors(e.g., false positives/false negatives). To reduce prediction errors, in this paper, we consider contextual information with access request elements to train up the learning models. The above example shows that *the contextual information of an access request might impact the users' decision.* Here, contextual information is a broad term that can be more formally represented as a tuple CTX of attributes, each one collecting a meaningful data to represent access request circumstances, like time of the request, current location of PDS's owner, etc. The aim of this work is to design a CP-PDS that, in presence of information about access request context, is able promptly react so as to have a decision that takes into account also users' preferences w.r.t. the context. By considering contextual data with access request elements, we can learn user privacy preferences by extending the framework presented in Chapter 5 with contextual data. More particularly, we can train the classifiers considering not only the access request elements but also contextual information. Precisely, we can exploit the ensemble approach as done in the Chapter 5 by having multiple classifiers built on access request elements complemented with contextual data CTX.

Figure 6.1 presents overall architecture of CP-PDS. According to our assumption, user contextual data combined with access request elements so as to get user opinions on upcoming access request. To do so, as shown in Figure 6.1, we first ask PDS owner about contextual based access request AR_CTX for labeling (cfr. Figure 6.1.a). According to our proposed learning model presented in Chapter 5, we exploit history-based active learning for selecting the most uncertain access request AR_CTX to be labeled by PDS owners, so as to build classifiers with less number of good quality of labeled training dataset (See labeling AR_CTX part in Figure 6.1). Since we exploit the ensemble approach by considering the contextual data with access request AR_CTX, thus we consider the different dimensions of access request elements with contextual data to train up multiple classifiers. More particularly, let assume that we build a classifier by considering contextual data CTX with (d_0,DC), another classifier can be built with contextual data CTX and (d_0,p). Like this way we can build multiple classifiers as what we did in Chapter 4 for building ensemble learning. The only extension in this chapter is that we consider contextual data with access request elements. Once the classifiers are built then the PHBAL can predict the class label on upcoming access requests (cfr. Figure 6.1.b).

6.3 Experiments

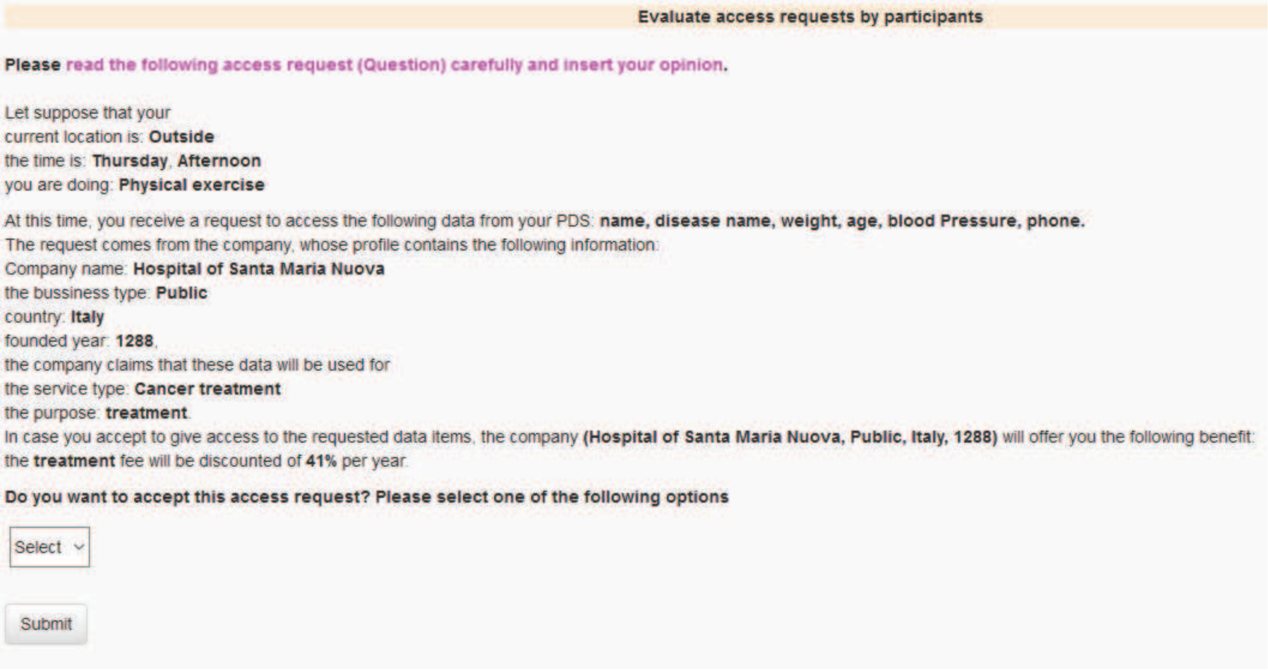
In this section, we illustrate the experiments we have performed to validate the proposed approach. More precisely, Section 6.3.3 presents a comparison of the proposed approach presented in Chapter 5 with the aim of assessing the importance of considering contextual information with access request. More particularly, we conduct this experiment to show the privacy preference accuracy of learning model when we consider non-contextual and contextual based access request.

Section 6.3.4, we perform the experiment that shows the accuracy on testing dataset, considering the contextual based access request. Moreover, in this experiment, we also explore the mechanism to reduce the over-fitting problem occurs in machine learning approaches. In general, over-fitting occurs when a learning model learns the noise/randomness along with the samples in the training dataset that negatively impacts the performance of the learning model on the upcoming new samples. For reducing the over-fitting, we need to do validation test with the learning models on training dataset with different approaches, that is, sequential and least probability.

In Section 6.3.5, we illustrate the experiment that shows how much user's decision are impacted with contextual data. Finally, in Section 6.3.6, we present the results about the accuracy on the testing dataset to check whether it will be increased or decreased based user quality in terms of feedback on the training dataset.

6.3.1 Experimental settings

Datasets. We collected two datasets: one dataset, referred to in what follows as DS-1, contains users' feedback on both non-contextual and contextual access requests separately, and another dataset, named DS-2, that contains users' feedback on contextual access requests only, as shown in Table 6.2. In DS-1, we only consider 20 users participation for investigating on which type of access requests (e.g., non-contextual/contextual access requests) users provide consistent feedbacks so as to check the performance of the learning model. Afterward, we consider more users involvement in DS-2 to check the performance of the learning model. We generate access requests containing realistic values for the data consumer, service type, requested data, purpose and offer fields. Moreover, we also consider contextual information, i.e., location, time, and activity. More precisely, we have considered: 55 different data consumer profiles; 18 different service types; 42 possible data fields;



Evaluate access requests by participants

Please read the following access request (Question) carefully and insert your opinion.

Let suppose that your
current location is: **Outside**
the time is: **Thursday**, **Afternoon**
you are doing: **Physical exercise**

At this time, you receive a request to access the following data from your PDS: **name, disease name, weight, age, blood Pressure, phone.**
The request comes from the company, whose profile contains the following information:
Company name: **Hospital of Santa Maria Nuova**
the bussiness type: **Public**
country: **Italy**
founded year: **1288**,
the company claims that these data will be used for
the service type: **Cancer treatment**
the purpose: **treatment**.
In case you accept to give access to the requested data items, the company **(Hospital of Santa Maria Nuova, Public, Italy, 1288)** will offer you the following benefit:
the **treatment** fee will be discounted of **41%** per year.

Do you want to accept this access request? Please select one of the following options

Figure 6.2: A sample of a contextual based access request to PDS

21 purposes; offer values ranging from 0% to 100%, and the following contextual data: 3 different locations (i.e., home, office, and outside), 7 days (e.g., Sunday, Monday. etc.), 4 time slots (i.e., morning, afternoon, evening, and night), and 13 different user's activities (e.g., meeting, driving, etc.). Based on these elements, we randomly generate access requests (see Figure 6.2 for an example). Since, we use semi-supervised learning, we need both labeled and unlabeled access requests. Each dataset contains 317 access requests. For dataset DS-1 , we ask labels for 20 non-contextual access requests and 20 contextual access requests, whereas for dataset DS-2, 60 access requests are labeled, whereas the remaining ones are used as unlabeled data.

Evaluators. For access request labeling, we developed a web application, and we use a crowdsourcing platform for user engagement. We have recruited 125 participants from the Microworker crowdsourcing platform[1] of different nationalities, ages, and educational levels. For dataset DS-1, we recruit 25 users to label 20 non-contextual and 20 contextual access requests. We exploit this dataset for checking the impact of contextual information in access decisions (see Sections6.3.3, and 6.3.5). For dataset DS-2 we recruit 100 workers, and we used it for the experiments in Sections 6.3.4, and 6.3.6.

To ensure good quality of the jobs submitted to Microworker, we have selected only workers with the best rating according to the Microworker platform. As further quality check, we measured the time each participant devoted to the labeling task and, if this is less than a reasonable time, we remove the participant. For dataset DS-2, we have presented

[1] https://www.microworkers.com

Table 6.2: Datasets used in the experiments

Dataset	User's opinions on contextual access requests	User's opinions on non-contextual access requests	# Labeled access requests	# Users
DS-1	✓	✓	20+20	25
DS-2	✓		60	100

72 access requests to each participant. More particularly, 60 access requests have been used as labeled training dataset; 7 access requests are used for testing the performance of the proposed approach, whereas 5 access requests are used for checking the quality of the job execution (see Section 6.3.6 for more details). The same approach has been applied to dataset DS-1, except for the number of requested labels. More precisely, we have presented 52 access requests to each participant: 20 non-contextual access requests, 20 contextual access requests, 7 access requests for testing purpose, whereas 5 access requests are used for checking the quality of user feedbacks.

6.3.2 Evaluation metrics

In this chapter, since we consider classes with 3 labels (yes, no, maybe) user can give opinion on contextual based access request, thus we exploit a 3X3 confusion matrix. More particularly, in order to measure the effectiveness of the proposed approach, we use the same traditional confusion matrix presented in Table 4.2 to define the evaluation metric, *accuracy* as the ratio of total number of true positives (TPs) to total number of samples.

6.3.3 Non-contextual vs contextual based access requests

In this experiment, we compare the accuracy of the learning model when we consider contextual based access request and non-contextual based access request. In this experiment, we exploit DS-1 dataset. Figure 6.3 shows the results on the testing dataset. Moreover, the experiment shown in Figure 6.3 confirms that the learning approach produces good accuracy on testing dataset when we consider contextual based access request than non-contextual based access request. It implies that users feedbacks on contextual based access requests are more consistent than on non-contextual based access requests.

6.3.4 Contextual based learning

In this experiment, we experiment our learning approach that exploit context and non-contextual information together. As shown in Figure 6.4, the accuracy on the testing dataset is around 75% produced by PHBAL. Moreover, since we use machine learning approaches to learn user privacy preference decisions, thus we need to consider strategies

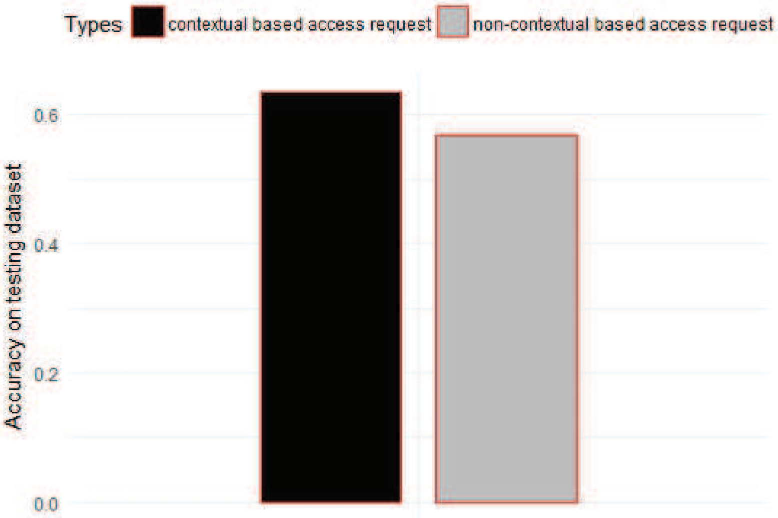


Figure 6.3: Accuracy on testing dataset compared between non-contextual and contextual based access requests

to minimize the prediction errors. As we know, machine learning has pitfall such as over-fitting [25, 103]. To deal with this problem, we select samples by varying the size of the training dataset, to check which one produces better accuracy on testing dataset. Prior researches have defined different approaches to deal with this issue [5, 57, 90]. Therefore, we experiment the following approaches, the first is the sequential based approach, which is a traditional approach to reduce the over-fitting problem in machine learning [62, 90]. This approach sequentially considers a set of labeled datasets of increasing size, and check the accuracy on the testing dataset. However, we have also considered an alternative approach, that we called least probability based approach. With this approach, we select the good quality labeled training dataset for learning models. To do so, we select the labeled training dataset which has least probability distance among classes from the labeled training pool and increasing the size based on the probability distance, and check the accuracy on the testing dataset. The experiment shows that least probability approach produces better accuracy on testing dataset than the sequential approach.

6.3.5 Impact of context on users' decisions

In this experiment, we investigate in how many access control decisions users have changed due to the consideration of contextual information. The experimental result reported in Figure 6.3 has already shown that a better accuracy is achieved when we consider contextual based access requests. In this experiment, we want to show how many access control decisions are driven by context data. For doing so, we first ask user opinions on non-

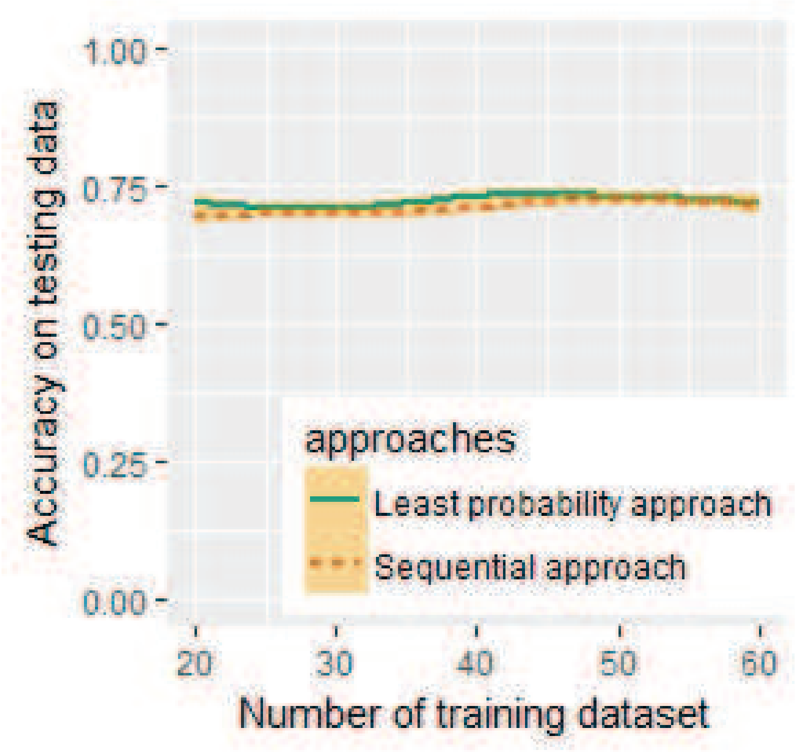


Figure 6.4: Accuracy on testing dataset

contextual access requests and then we ask again the user feedback on the same access request with the associated contextual information. Figure 6.5 shows the results. More particularly, the experiment shows that 66.87% decisions are the same, but, interestingly, 33.13% of the access request decisions are changed by users when we consider contextual information.

6.3.6 Participants quality

Clearly, the output of any machine learning approach depends on the quality of the user's input on the training dataset. Therefore, we are interested to investigate how a badly labeled training dataset impacts the final decision on the testing dataset. Thus, we set some strategies to identify consistent and inconsistent evaluators. First, three of the access requests are presented twice to evaluators, to check if they are always marked with the same label. Based on the assigned labels, we can judge whether the evaluator is consistent or not in his/her decisions, which gives us a measure of the quality of his/her jobs. Second, we have inserted two access requests in the first phase (e.g., among the first nine access requests) which contain a requested data field which is inconsistent with the requested access purpose and service. For example, we ask a label for an access request on these data {*traveling date, traveling time, From (starting place), To (destination place), etc.,*},

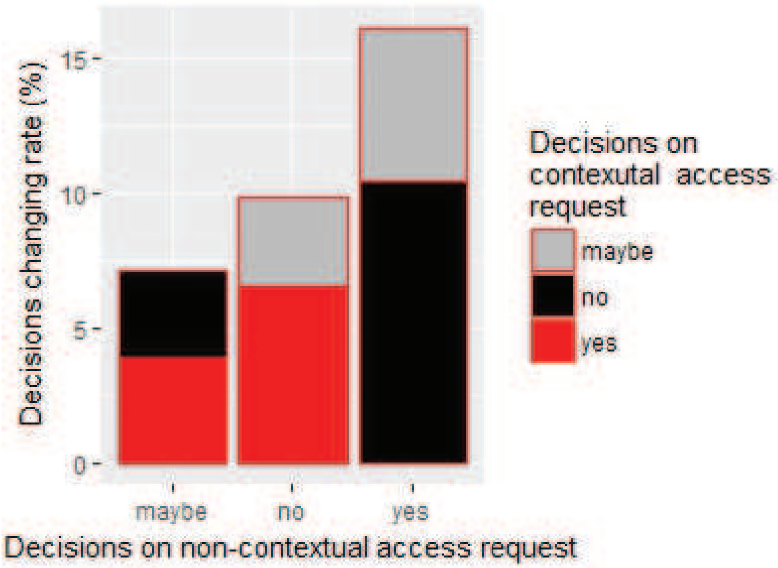


Figure 6.5: Users' decisions changing rate based on contextual based access requests

having a service purpose *issuing a loan*. We expect that, in case of an inconsistent access request, a participant that carefully reads the request will assign a deny label. Therefore, we consider a participant as consistent if he/she behaves correctly w.r.t. the above described checks. In our experiment, we show that 43% users have given feedback on access requests in consistent manner, whereas 57% users are inconsistent.

Figure 6.6 presents the comparative analysis of the accuracy level for consistent and inconsistent participants on the testing dataset for the learning approach PHBAL that exploits the contextual data with access request elements for learning user privacy preferences. The figure shows that the accuracy achieved with consistent users is greater than the one achieved from inconsistent users. It confirms that good quality labeled training dataset can improve the performance of the learning model. The experiment also shows that least probability approach produces better accuracy on testing dataset than the sequential approach.

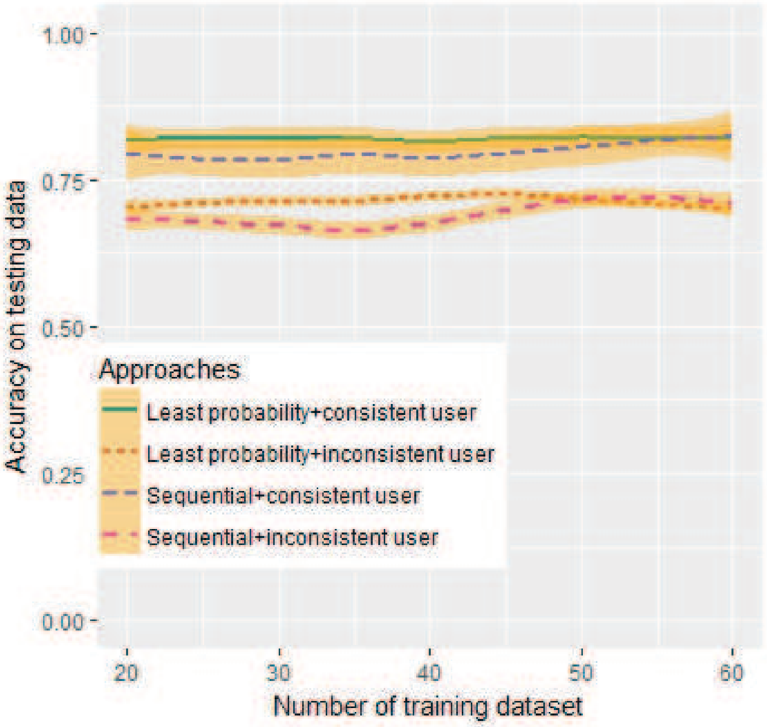


Figure 6.6: Accuracy on testing dataset for consistent and inconsistent users

6.4 Chapter summary

In this chapter, we simply extent the approach presented in Chapter 5 by considering user contextual data. The experimental results show that the proposed privacy preference framework produces better result when we consider user contextual data in term of getting user feedbacks on access requests. Moreover, machine learning has pitfall such as over-fitting [25, 103]. In general, over-fitting occurs when a learning model learns the noise/randomness along with the samples in the training dataset that negatively impacts the performance of the learning model on the upcoming new samples. For reducing the over-fitting, we do validation test with the learning models on training dataset with an approach called least probability approach proposed in this chapter provides better result than traditional sequential approach.

Chapter 7

Conclusion and Future work

Personal Data Storage (PDS) enables individuals to store their personal data into a unique digital repository so as users can have full control on their data. In fact, PDS offers individuals the capability to keep their data into a unique logical repository, where they can be connected and exploited by proper analytical tools, as well as shared with third parties under the control of end users. However, several research has shown that average users are not so skillful to define their privacy preferences properly. Therefore in this thesis, we have addressed different privacy preference mechanisms for PDS so as to help user's to manage their privacy preferences in PDS. In Chapter 3, we has discussed a new approach for the design of a privacy-aware Personal Data Store (PDS) based on risk-benefit assessment. In this chapter, we proposed the architecture of PDS as well as a suite of strategies to share personal data with data consumers. In Chapter 4, we have considered the issue of learning privacy habits of PDS owners. This is a crucial aspect since it may help in reducing the burden of privacy preference specification. We have considered different learning approaches to test which one better performs in the considered scenario. Such approaches allow us to focus on different aspects of the user access request decisions. We have extensively tested our approaches by using evaluators enrolled from the university environment, as well as through a crowd-sourcing platform. The achieved experimental results are promising. However, it is obvious that machine learning approaches exploit user's feedbacks to train up the learning models for predicting user's opinions on the upcoming instances. Thus, good quality training dataset can improve the performance of the learning models. Therefore, in Chapter 5 we have proposed a Privacy-aware Personal Data Storage, able to automatically take privacy-aware decisions on third parties access requests in accordance with user preferences based on semi-supervised and active learning approaches. Moreover, the system relies on active learning complemented with strategies to strengthen user privacy protection. As discussed in the chapter, we run several experiments on a realistic dataset and exploiting a group of 360 evaluators. The obtained results show the effectiveness of the proposed approach. In Chapter 6, we extent the approach presented in Chapter 5 by considering user contextual data. The obtained results show that contextual data play a essential role in term of implementing privacy preference models in PDS.

We plan to extend our works along several directions. A first direction is the investigation of several functions to compute the risk and benefit values, taking into account different dimensions of the data owners' perspective in Chapter 3. Furthermore, we plan to implement a prototype of our privacy-aware PDS and test it in different real world scenarios. Then, we plan to extent the Chapter 4 to define a mechanism to exploit the obtained classifiers for suggesting customized privacy preferences, that is, a set of rules defined based on the PDS owner privacy aptitudes in access request evaluation. Moreover, we plan to include strategies on support of privacy preference modification for misconfiguration cases, based on the history of access requests made by service providers. This might happen for those users who do not maintain consistency in terms of the given feedback on access requests. Furthermore, we plan to conduct more user studies for understanding how users interact with our framework and comparing it to alternative privacy settings approaches. Next, we plan to extend the Chapter 5 along several directions. First, we are interested to investigate how P-PDS could scale in the IoT scenario, where access requests decision might depend also on contexts, not only on user preferences. Also, we would like to integrate P-PDS with cloud computing services (e.g., storage and computing) so as to design a more powerful P-PDS by, at the same time, protecting users' privacy. Finally, our future plan is to extent the Chapter 6 by proposing a solution that can define polices to compute similarity aptitude of PDS owners so as to build privacy preferences models by exploiting these similarity features.

Bibliography

[1] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[2] Cuneyt Akcora, Barbara Carminati, and Elena Ferrari. Privacy in social networks: How risky is your social graph? In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 9–19. IEEE, 2012.

[3] Davide Alberto Albertini, Barbara Carminati, and Elena Ferrari. Privacy settings recommender for online social network. In *Collaboration and Internet Computing (CIC), 2016 IEEE 2nd International Conference on*, pages 514–521. IEEE, 2016.

[4] Jan Philipp Albrecht. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[5] Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *AAAI*, pages 1673–1679, 2014.

[6] Tristan Allard, Nicolas Anciaux, Luc Bouganim, Yanli Guo, Lionel Le Folgoc, Benjamin Nguyen, Philippe Pucheral, Indrajit Ray, Indrakshi Ray, and Shaoyi Yin. Secure personal data servers: a vision paper. *Proceedings of the VLDB Endowment*, 3(1-2):25–35, 2010.

[7] Michael Backes, Aniket Kate, Matteo Maffei, and Kim Pecina. Obliviad: Provably secure and practical online behavioral advertising. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 257–271. IEEE, 2012.

[8] Paritosh Bahirat, Yangyang He, Abhilash Menon, and Bart Knijnenburg. A data-driven approach to developing iot privacy-setting interfaces. In *23rd International Conference on Intelligent User Interfaces*, pages 165–176. ACM, 2018.

[9] Gordon Bell. A personal digital store. *Communications of the ACM*, 44(1):86–91, 2001.

[10] Elisa Bertino, Carolyn Brodie, Seraphin B Calo, Lorrie Faith Cranor, C Karat, John Karat, Ninghui Li, Dan Lin, Jorge Lobo, Qun Ni, et al. Analysis of privacy and security policies. *IBM Journal of Research and Development*, 53(2):3–1, 2009.

[11] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

[12] Greg Bigwood, F Ben Abdesslem, and Tristan Henderson. Predicting location-sharing privacy preferences in social network applications. *Proc. of AwareCast*, 12:1–12, 2012.

[13] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[14] Igor Bilogrevic, Kévin Huguenin, Berker Agir, Murtuza Jadliwala, Maria Gazaki, and Jean-Pierre Hubaux. A machine-learning based approach to privacy-aware information-sharing in mobile social networks. *Pervasive and Mobile Computing*, 25:125–142, 2016.

[15] Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M Jones. mlr: Machine learning in r. *Journal of Machine Learning Research*, 17(170):1–5, 2016.

[16] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[17] Kathy Bohrer, Xuan Liu, Dogan Kesdogan, Edith Schonberg, Moninder Singh, and Susan Spraragen. Personal information management and distribution. In *4th International Conference on Electronic Commerce Research ICECR-4*, 2001.

[18] Joseph Bonneau and Sören Preibusch. The privacy jungle: On the market for data protection in social networks. In *Economics of information security and privacy*, pages 121–167. Springer, 2010.

[19] Sean Borman. The expectation maximization algorithm-a short tutorial. *Submitted for publication*, pages 1–9, 2004.

[20] Mohamed-Rafik Bouguelia, Yolande Belaïd, and Abdel Belaïd. A stream-based semi-supervised active learning approach for document classification. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 611–615. IEEE, 2013.

[21] Paul S Bradley, Usama Fayyad, Cory Reina, et al. Scaling em (expectation-maximization) clustering to large databases. Technical report, Technical Report MSR-TR-98-35, Microsoft Research Redmond, 1998.

[22] Ulf Brefeld. Multi-view learning with dependent views. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 865–870. ACM, 2015.

[23] Carl Burch. A survey of machine learning. *A survey for the Pennsylvania Governor's School for the Sciences*, 2001.

[24] Ji-Won Byun and Ninghui Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal—The International Journal on Very Large Data Bases*, 17(4):603–619, 2008.

[25] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

[26] Tej Chajed, Jon Gjengset, M Frans Kaashoek, James Mickens, Robert Morris, and Nickolai Zeldovich. Oort: User-centric cloud storage with global queries. 2016.

[27] Tej Chajed, Jon Gjengset, Jelle Van Den Hooff, M Frans Kaashoek, James Mickens, Robert Morris, and Nickolai Zeldovich. Amber: Decoupling user data from web applications. In *HotOS*, volume 15, pages 1–6, 2015.

[28] Supriyo Chakraborty, Chenguang Shen, Kasturi Rangan Raghavan, Yasser Shoukry, Matt Millar, and Mani B Srivastava. ipshield: A framework for enforcing context-aware privacy. In *NSDI*, pages 143–156, 2014.

[29] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[30] Yuan Cheng, Jaehong Park, and Ravi Sandhu. Preserving user privacy from third-party applications in online social networks. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 723–728. ACM, 2013.

[31] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[32] MacKenzie F Common. Facebook and cambridge analytica: let this be the high-water mark for impunity. *LSE Business Review*, 2018.

[33] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12), 2009.

[34] George Danezis. Inferring privacy policies for social networking services. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, pages 5–10. ACM, 2009.

[35] Yves-Alexandre de Montjoye, Erez Shmueli, Samuel S Wang, and Alex Sandy Pentland. openpds: Protecting the privacy of metadata through safeanswers. *PloS one*, 9(7):e98790, 2014.

[36] Thomas G Dietterich. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002.

[37] Thomas G Dietterich et al. Ensemble methods in machine learning. *Multiple classifier systems*, 1857:1–15, 2000.

[38] Catherine Dwyer, Starr Hiltz, and Katia Passerini. Trust and privacy concern within social networking sites: A comparison of facebook and myspace. *AMCIS 2007 proceedings*, page 339, 2007.

[39] Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360. ACM, 2010.

[40] Joshua Fogel and Elham Nehmad. Internet social network communities: Risk taking, trust, and privacy concerns. *Computers in human behavior*, 25(1):153–160, 2009.

[41] Kambiz Ghazinour, Stan Matwin, and Marina Sokolova. Monitoring and recommending privacy settings in social networks. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 164–168. ACM, 2013.

[42] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[43] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *USENIX conference on Networked systems design and implementation*, pages 169–182, 2011.

[44] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.

[45] Hamza Harkous, Rameez Rahman, and Karl Aberer. C3p: Context-aware crowdsourced cloud privacy. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 102–122. Springer, 2014.

[46] Jason I Hong and James A Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189. ACM, 2004.

[47] Keith Irwin and Ting Yu. Determining user privacy preferences by asking the right questions: an automated approach. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 47–50. ACM, 2005.

[48] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[49] Qi Jia, Linke Guo, Zhanpeng Jin, and Yuguang Fang. Privacy-preserving data classification and similarity evaluation for distributed systems. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, pages 690–699. IEEE, 2016.

[50] Hemanth Khambhammettu, Sofiene Boulares, Kamel Adi, and Luigi Logrippo. A framework for risk assessment in access control systems. *computers & security*, 39:86–103, 2013.

[51] Jan Kolter and Günther Pernul. Generating user-understandable privacy preferences. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 299–306. IEEE, 2009.

[52] Andreas Krause and Eric Horvitz. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research*, 39:633–662, 2010.

[53] Naeimeh Laleh, Barbara Carminati, and Elena Ferrari. Risk assessment in social networks based on user anomalous behaviour. *IEEE Transactions on Dependable and Secure Computing*, 2016.

[54] Ryan Layfield, Murat Kantarcioglu, and Bhavani Thuraisingham. Incentive and trust issues in assured information sharing. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 113–125. Springer, 2008.

[55] Pedro Giovanni Leon, Blase Ur, Yang Wang, Manya Sleeper, Rebecca Balebako, Richard Shay, Lujo Bauer, Mihai Christodorescu, and Lorrie Faith Cranor. What matters to users?: factors that affect users' willingness to share information with online advertisers. In *Proceedings of the ninth symposium on usable privacy and security*, page 7. ACM, 2013.

[56] Tingting Liang, Lifang He, Chun-Ta Lu, Liang Chen, Philip S Yu, and Jian Wu. A broad learning approach for context-aware mobile application recommendation. *arXiv preprint arXiv:1709.03621*, 2017.

[57] Florian Liebgott and Bin Yang. Active learning with cross-dataset validation in event-based non-intrusive load monitoring. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 296–300. IEEE, 2017.

[58] Jialiu Lin, Bin Liu, Norman Sadeh, and Jason I Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. 2014.

[59] Heather Richter Lipford, Andrew Besmer, and Jason Watson. Understanding privacy settings in facebook with an audience view. *UPSEC*, 8:1–8, 2008.

[60] Bin Liu, Jialiu Lin, and Norman Sadeh. Reconciling mobile app privacy and usability on smartphones: Could user privacy profiles help? In *Proceedings of the 23rd international conference on World wide web*, pages 201–212. ACM, 2014.

[61] Yabing Liu, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 61–70. ACM, 2011.

[62] David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[63] Michelle Madejski, Maritza Johnson, and Steven M Bellovin. A study of privacy settings errors in an online social network. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 340–345. IEEE, 2012.

[64] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, et al. Owl-s: Semantic markup for web services. *W3C member submission*, 22:2007–04, 2004.

[65] David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burnstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry R Payne, Marta Sabou, Monika Solanki, et al. Bringing semantics to web services: The owl-s approach. 2004.

[66] Alessandra Mazzia, Kristen LeFevre, and Eytan Adar. The pviz comprehension tool for social network privacy settings. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 13. ACM, 2012.

[67] Andrew Kachites McCallumzy and Kamal Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

[68] Saif M Mohammad and Graeme Hirst. Distributional measures of semantic distance: A survey. *arXiv preprint arXiv:1203.1858*, 2012.

[69] Marco Casassa Mont and Robert Thyne. A systemic approach to automate privacy policy enforcement in enterprises. In *International Workshop on Privacy Enhancing Technologies*, pages 118–134. Springer, 2006.

[70] Richard Mortier, Jianxin Zhao, Jon Crowcroft, Qi Li, Liang Wang, Hamed Haddadi, Yousef Amar, Andy Crabtree, James Colley, Tom Lodge, et al. Personal data management with the databox: what's inside the box? 2016.

[71] Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal data vaults: a locus of control for personal data streams. In *Proceedings of the 6th International COnference*, page 17. ACM, 2010.

[72] Ion Muslea, Steven Minton, and Craig A Knoblock. Active+ semi-supervised learning= robust multi-view learning. In *ICML*, volume 2, pages 435–442, 2002.

[73] Pardis Emami Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Cranor, and Norman Sadeh. Privacy expectations and preferences in an iot world. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[74] Divya G Nair, VP Binu, and G Santhosh Kumar. An effective private data storage and retrieval system using secret sharing scheme based on secure multi-party computation. *arXiv preprint arXiv:1502.07994*, 2015.

[75] Toru Nakamura, Shinsaku Kiyomoto, Welderufael B Tesfay, and Jetzabel Serna. Easing the burden of setting privacy preferences: A machine learning approach. In *International Conference on Information Systems Security and Privacy*, pages 44–63. Springer, 2016.

[76] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

[77] Helen Nissenbaum. A contextual approach to privacy online. *Daedalus*, 140(4):32–48, 2011.

[78] Katarzyna Olejnik, Italo Dacosta, Joana Soares Machado, Kévin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. Smarper: Context-aware and automatic runtime-permissions for mobile devices. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 1058–1076. IEEE, 2017.

[79] Robi Polikar. Ensemble learning. In *Ensemble machine learning*, pages 1–34. Springer, 2012.

[80] Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker, and Jinghai Rao. Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing*, 13(6):401–412, 2009.

[81] Roman Schlegel, Apu Kapadia, and Adam J Lee. Eyeing your exposure: quantifying and controlling information sharing for improved privacy. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, page 14. ACM, 2011.

[82] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[83] Riaz Ahmed Shaikh, Kamel Adi, and Luigi Logrippo. Dynamic risk-based decision methods for access control systems. *computers & security*, 31(4):447–464, 2012.

[84] Bikash Chandra Singh, Barbara Carminati, and Elena Ferrari. A risk-benefit driven architecture for personal data release. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*, pages 40–49. IEEE, 2016.

[85] Bikash Chandra Singh, Barbara Carminati, and Elena Ferrari. Learning privacy habits of pds owners. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 151–161. IEEE, 2017.

[86] Kapil Singh, Sumeer Bhola, and Wenke Lee. xbook: Redesigning privacy control in social networking platforms. In *USENIX Security Symposium*, pages 249–266, 2009.

[87] Ian Smith, Sunny Consolvo, Anthony Lamarca, Jeffrey Hightower, James Scott, Timothy Sohn, Jeff Hughes, Giovanni Iachello, and Gregory D Abowd. Social disclosure of place: From location technology to communication practices. In *International Conference on Pervasive Computing*, pages 134–151. Springer, 2005.

[88] Ana Stanescu and Doina Caragea. Ensemble-based semi-supervised learning approaches for imbalanced splice site datasets. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 432–437. IEEE, 2014.

[89] Katherine Strater and Heather Richter Lipford. Strategies and struggles with privacy in an online social networking community. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, pages 111–119. British Computer Society, 2008.

[90] Masashi Sugiyama and Neil Rubens. A batch ensemble approach to active learning with model selection. *Neural Networks*, 21(9):1278–1286, 2008.

[91] Brian Michael Sweatt et al. *A privacy-preserving personal sensor data ecosystem.* PhD thesis, Massachusetts Institute of Technology, 2014.

[92] Brian Tarran. What can we learn from the facebook—cambridge analytica scandal? *Significance*, 15(3):4–5, 2018.

[93] Mohit Tiwari, Prashanth Mohan, Andrew Osheroff, Hilfi Alkaff, Elaine Shi, Eric Love, Dawn Song, and Krste Asanović. Context-centric security. In *Proceedings of the 7th USENIX conference on Hot Topics in Security*, pages 9–9. USENIX Association, 2012.

[94] Eran Toch. Crowdsourcing privacy preferences in context-aware applications. *Personal and ubiquitous computing*, 18(1):129–141, 2014.

[95] Eran Toch, Justin Cranshaw, Paul Hankes Drielsma, Janice Y Tsai, Patrick Gage Kelley, James Springfield, Lorrie Cranor, Jason Hong, and Norman Sadeh. Empirical models of privacy in location sharing. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 129–138. ACM, 2010.

[96] Lynn Tsai, Primal Wijesekera, Joel Reardon, Irwin Reyes, Serge Egelman, David Wagner, Nathan Good, and Jung-Wei Chen. Turtle guard: Helping android users apply contextual privacy preferences. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[97] Bimal Viswanath, Emre Kiciman, and Stefan Saroiu. Keeping information safe from social networking apps. In *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pages 49–54. ACM, 2012.

[98] Frank Wang, James Mickens, Nickolai Zeldovich, and Vinod Vaikuntanathan. Sieve: Cryptographically enforced access control for user data in untrusted clouds. In *NSDI*, pages 611–626, 2016.

[99] Qihua Wang and Hongxia Jin. Quantified risk-adaptive access control for patient privacy protection in health information systems. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 406–410. ACM, 2011.

[100] Yan Wang, Mooi-Choo Chuah, and Yingying Chen. Incentive driven information sharing in delay tolerant mobile networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 5279–5284. IEEE, 2012.

[101] Royu Want, Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, and John Light. The personal server: Changing the way we think about ubiquitous computing. In *International Conference on Ubiquitous Computing*, pages 194–209. Springer, 2002.

[102] Alan F Westin. Social and political dimensions of privacy. *Journal of social issues*, 59(2):431–453, 2003.

[103] Shimon Whiteson, Brian Tanner, Matthew E Taylor, and Peter Stone. Protecting against evaluation overfitting in empirical reinforcement learning. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 120–127. IEEE, 2011.

[104] Jason Wiese, Patrick Gage Kelley, Lorrie Faith Cranor, Laura Dabbish, Jason I Hong, and John Zimmerman. Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 197–206. ACM, 2011.

[105] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. Android permissions remystified: A field study on contextual integrity. In *USENIX Security Symposium*, pages 499–514, 2015.

[106] Primal Wijesekera, Joel Reardon, Irwin Reyes, Lynn Tsai, Jung-Wei Chen, Nathan Good, David Wagner, Konstantin Beznosov, and Serge Egelman. Contextualizing

privacy decisions for better prediction (and protection). In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 268. ACM, 2018.

[107] Hongchen Wu, Bart P Knijnenburg, and Alfred Kobsa. Improving the prediction of users' disclosure behavior by making them disclose more predictably? In *Symposium on Usable Privacy and Security (SOUPS)*, 2014.

[108] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 229–240. ACM, 2006.

[109] Jierui Xie, Bart Piet Knijnenburg, and Hongxia Jin. Location sharing privacy preference: analysis and personalized recommendation. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pages 189–198. ACM, 2014.

[110] Qiang Xiong and Xiaoyan Chen. Incentive mechanism design based on repeated game theory in security information sharing. In *2nd International Conference on Science and Social Research (ICSSR 2013). Atlantis Press*, 2013.

[111] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[112] Kaihe Xu, Haichuan Ding, Linke Guo, and Yuguang Fang. A secure collaborative machine learning framework based on data locality. In *Global Communications Conference (GLOBECOM), 2015 IEEE*, pages 1–5. IEEE, 2015.

[113] Kaihe Xu, Hao Yue, Linke Guo, Yuanxiong Guo, and Yuguang Fang. Privacy-preserving machine learning algorithms for big data systems. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 318–327. IEEE, 2015.

[114] Zhu Yan, Guhua Gan, and Khaled Riad. Bc-pds: Protecting privacy and self-sovereignty through blockchains for openpds. In *Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on*, pages 138–144. IEEE, 2017.

[115] Xing Yi, Yunpeng Xu, and Changshui Zhang. Multi-view em algorithm for finite mixture models. In *International Conference on Pattern Recognition and Image Analysis*, pages 420–425. Springer, 2005.

[116] Lin Yuan, Joël Theytaz, and Touradj Ebrahimi. Context-dependent privacy-aware photo sharing based on machine learning. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 93–107. Springer, 2017.

[117] Yuchen Zhao. *Recommending privacy preferences in location-sharing services.* PhD thesis, University of St Andrews, 2017.

[118] Zhi-Hua Zhou. When semi-supervised learning meets ensemble learning. In *International Workshop on Multiple Classifier Systems*, pages 529–538. Springer, 2009.

[119] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.

[120] Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi. Active learning from data streams. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 757–762. IEEE, 2007.

[121] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes. Active learning with drifting streaming data. *IEEE transactions on neural networks and learning systems*, 25(1):27–39, 2014.

Appendices

Appendix A

Publications

A.1 International Conferences

1. Bikash Chandra Singh, Barbara Carminati, and Elena Ferrari, "A Risk-Benefit Driven Architecture for Personal Data Release" in IEEE International Conference on Information Reuse and Integration, 2016.

   Abstract: Personal data storages (PDSs) give individuals the ability to store their personal data in a data unified repository and control release of their data to data consumers. Being able to gather personal data from different data sources (e.g., banks, hospitals), PDSs will play strategic role in individual privacy management. As such, PDS demands for new privacy models for protecting personal data. In this paper, we propose a new technical approach that empowers individuals to better control data in PDS. Particularly, we present a privacy-aware PDS architecture by focusing on two logical data zones based on the categories of personal data. Moreover, we propose a strategy for regulating personal data release that takes in consideration both user preferences and possible risks and benefits of the data release.

2. Bikash Chandra Singh, Barbara Carminati, and Elena Ferrari, "Learning privacy habits of PDS owners" in IEEE International Conference on Distributed Computing Systems (ICDCS), 2017.

   Abstract: The concept of Personal Data Storage (PDS) has recently emerged as an alternative and innovative way of managing personal data w.r.t. the service-centric one commonly used today. The PDS offers a unique logical repository, allowing individuals to collect, store, and give access to their data to third parties. The research on PDS has so far mainly focused on the enforcement mechanisms, that is, on how user privacy preferences can be enforced. In contrast, the fundamental issue of preference specification has been so far not deeply investigated. In this paper, we do a step in this direction by proposing different learning algorithms that allow a fine-grained learning of the privacy aptitudes of PDS owners. The learned models are then used to answer third party access requests. The extensive experiments we have performed show the effectiveness of the proposed approach.

A.2 International Journals

1. Bikash Chandra Singh, Barbara Carminati, and Elena Ferrari, “Privacy-aware Personal Data Storage (P-PDS): Protecting User Privacy from External Applications”, submitted to IEEE Transactions on Dependable and Secure Computing (TDSC).

   Abstract: Recently, Personal Data Storage (PDS) has inaugurated a substantial change to the way people can store and control their personal data, by moving from a service-centric to a user-centric model. PDS offers individuals the capability to keep their data into a unique logical repository, that can be connected and exploited by proper analytical tools, or shared with third parties under the control of end users. Up to now, most of the research on PDS has focused on how to enforce user privacy preferences and how to secure data when stored into the PDS. In contrast, in this paper we aim at designing a *Privacy-aware Personal Data Storage* (P-PDS), that is, a PDS able to automatically take privacy-aware decisions on third parties access requests in accordance with user preferences. The proposed P-PDS is based on preliminary results presented in [85], where it has been demonstrated that semi-supervised learning can be successfully exploited to make a PDS able to automatically decide whether an access request has to be authorized or not. In this paper, we have deeply revised the learning process so as to have a more usable P-PDS, in terms of reduced effort for the training phase, as well as a more conservative approach w.r.t. users’ privacy, when handling conflicting access requests. We run several experiments on a realistic dataset and exploiting a group of 360 evaluators. The obtained results show the effectiveness of the proposed approach.