Privacy Aware Machine Learning

A thesis submitted to Indian Statistical Institute

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

by

Chandan Biswas

Senior Research Fellow

Under the supervision of

Dr. Ujjwal Bhattacharya and Dr. Debasis Ganguly

Computer Vision and Pattern Recognition Unit

Indian Statistical Institute, Kolkata

August 2023

To Everyone who believed that I could do this...

Acknowledgements

At the onset, I would like to express my profound sense of gratitude to both of my supervisors Dr. Ujjwal Bhattacharya and Dr.Debasis Ganguly for introducing me to this research topic and providing their valuable guidance and unfailing encouragement throughout the entire period of my PhD work. Their deep insight and all-round support not only helped me to complete the work for this thesis in a timely manner but also they had set noble examples to follow in my future career. Also, I am grateful to Prof. Mandar Mitra and Prof. Utpal Garain of CVPR Unit of Indian Statistical Institute, who helped me by providing thoughtful suggestions on various administrative matters during the whole period of my stay at the Institute as a research scholar. I acknowledge all other faculty members of Computer Vision and Pattern Recognition Unit of Indian Statistical Institute. I am also grateful to Mr. Manick Dutta and Mr. Sankar Sen of the Office of CVPR Unit for their care and support to me.

I express my big thanks to all of my colleagues who worked with our research group from time to time. I must name a few of them - Partha Sarathi Mukherjee, Dwaipayan Roy, Sudip Das, Chanchal Biswas, Abhisek Chakrabarty, Jayati Mukherjee among many others for their interests in my research works as well as their support during some of the oddest conversations in the tea breaks. Special thanks must go to Partha Sarathi Mukherjee, who always provided me insightful ideas, deep learning programming guidance, constant support and inspiration throughout the course of this thesis.

Words are not enough to thank my family members, specially my wife, parents and brother for their unconditional support to me throughout the ups and downs of the entire duration of my PhD study. Here are my special thanks for my wife Rita Biswas who provided me persistent support and acted as the source of my inspiration during the stressful moments throughout the period.

Last but certainly not the least, I am thankful to the authority of Indian Statistical Institute, Kolkata for providing me research fellowship to do my PhD work, a good environment as well as the infrastructure without which completion of this thesis could never be accomplished.

Abstract

Privacy preserving computation is of utmost importance in a cloud computing environment where a client often requires to send sensitive data to servers, offering computing services, for computational purposes over untrusted networks. Sharing the raw or an abstract representation of a labelled or unlabelled dataset on cloud platforms can potentially expose sensitive information of the data to an adversary, e.g., in the case of an emotion classification task from text, an adversary-agnostic abstract representation of the text data may eventually lead an adversary to identify the demographics of the authors, such as their gender and age, etc. The leakage of sensitive information from the data may take place due to eavesdropping over the network or malware residing at the server. Privacy preserving computation workflows aim to prevent such leakage of sensitive information by introducing a suitable encoding transformation on sample data points. Such an encoding strategy has dual objectives, the first being that it should be difficult to reconstruct the original data in the absence of any knowledge of the encoding strategy and its parameters. Secondly, the computational results obtained using the encoded data should not be substantially different from those obtained using the same data in its original form. Standard encoding mechanisms, such as locality sensitive hashing (LSH), caters to the first objective of privacy preserving computation workflow, the second objective may not always be adequately satisfied. In this thesis, we focus on the second objective and the computational activity that we focus on is a supervised classification task in addition to the K-means clustering, which has been widely used for various data mining jobs. Here, we have addressed the problem of privacy preserving computation on the above two tasks in three different ways,

Initially, we have proposed a new variant of the K-means algorithm which is capable of privacy preservation in the sense that it takes binary encoded data as input, and does not require access to the data in its original form at any stage of the computation. The proposed strategy is capable of producing the required number of clusters which are sufficiently close to the respective clusters computed from the original non-encoded data. The results of the proposed strategy on image or text data are either comparable or outperform the standard K-means clustering algorithm.

Secondly, we have explored a deep metric learning approach to learn a parameterized encoding transformation with an objective of maximizing the alignment of the clusters obtained in the encoded space with the same obtained from the original data. To this end, we train a weakly supervised deep network using triplets constructed from the output of a clustering algorithm on a subset of the non-encoded data. Our proposed method of weakly-supervised approach yields more effective encoding in comparison to approaches where the encoding process is agnostic of the clustering objective.

Finally, we propose a universal defense mechanism against malicious attempts of stealing sensitive information from data shared on cloud platforms. More specifically, our proposed method employs an informative subspace based multi-objective approach to produce a sensitive information aware encoding of the data representation. A number of experiments conducted on both standard text and image datasets demonstrate the ability of our proposed approach to reduce the effectiveness of the adversarial task without remarkably affecting the effectiveness of the primary task itself.

List of Publications by the Author Related to the Thesis

[1] C. Biswas, D. Ganguly, D. Roy, and U. Bhattacharya. Privacy preserving approximate k-means clustering. In *Proc. of the 28th ACM International Conference on Information and Knowledge Management*, pages 1321–1330, 2019. doi: 10.1145/3357384.3357969. URL https://doi.org/10.1145/3357384.3357969.

[2] C. Biswas, D. Ganguly, and U. Bhattacharya. Approximate nearest neighbour search on privacy-aware encoding of user locations to identify susceptible infections in simulated epidemics. In *FIRE 2021: Forum for Information Retrieval Evaluation*, pages 35–42, 2021. doi: 10.1145/3503162.3503164. URL https://doi.org/10.1145/3503162.3503164.

[3] C. Biswas, D. Ganguly, P. S. Mukherjee, U. Bhattacharya, and Y. Hou. Privacy-aware supervised classification: An informative subspace based multi-objective approach. *Pattern Recognition*, 122:108301, 2022. doi: 10.1016/j.patcog.2021.108301. URL https://doi.org/10.1016/j.patcog.2021.108301.

[4] C. Biswas, D. Ganguly, D. Roy, and U. Bhattacharya. Weakly supervised deep metric learning on discrete metric spaces for privacy-preserved clustering. *Information Processing & Management*, pages 103109, 2023. doi: 10.1016/j.ipm.2022.103109. URL https://doi.org/10.1016/j.ipm.2022.103109.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Latest developments in the area of Machine Learning (ML) have shown great success across a wide range of application domains such as healthcare [106], banking [85], agriculture [70], business [78], transportation [149], and many others. Several of its efficient tools have been widely used to formulate predictive models for various applications such as image classification [26], text recognition [21], speech recognition [1], etc. Generally, the performance of the ML model depends on the volume of training data, large volume helps to capture the inherent characteristics or the distribution of the data.

Now a days many organizations are independently digitizing and storing the private data such as medical records, financial data etc. Often this type of private data implicitly or explicitly contain sensitive information. For example, consider the product review data, which may implicitly or explicitly contain the information corresponding to the age group or gender of the review writer, which are sensitive in nature. Now, the demographic information of the review writer can be predicted using the linguistic cues in the text itself [118, 113].

The most common practice of publishing a dataset that contains sensitive information is to anonymize the dataset. Unfortunately, number of past studies shown that removing only the type of information which are considered as 'personal' is not sufficient to protect the subject's identity, an adversary can re-identify and breach privacy of the individuals data records using cross-corelation with some other databases which may be publicly available. For example, the privacy breaches caused by infamous AOL *search query data scandal*, which de-anonymised the Massachusetts hospital discharge database by cross-corelating it with a public voter database [105]. Another well known example is the re-identification of user records from anonymised training dataset of *Netflix Prize* competition. In this competition the famous online DVD-rental and video streaming service company Netflix announced an "one million dollar" prize to be awarded for the best collaborative filtering algorithm to predict the user rating for films. They publicly release an anonymised training dataset of movie ratings. But within few days the authors of [108] shown that the user's record in the training data can be re-identify by matching the movie ratings from the Internet Movie Database, an open movie rating website. Many other work on re-identification of subject's records from anonymised dataset have been published in past literature which includes identifying participants records in the Personal Genome Project (PGP) by cross-referencing publicly available databases such as voter registration archive [139], re-identifying patient records in Washington state health records by looking up the information in hospitalization news paper stories [138], etc.

On the other hand, the use of the private data to train an ML model may raise serious privacy concerns due to the presence of sensitive information in the data. Consequently, development of privacy aware ML (from hereon, we interchangeably call as privacy-preserving ML) solutions is essential to take care of privacy concerns of the client data. It should be able to preserve the privacy of the data without compromising its effectiveness. The most common approaches of the development of this type of ML solutions include privacy aware encoding [34, 137, 131], differential privacy [16, 37], homomorphic encryption [87, 119], etc. In our present research we have used privacy aware encoding of the data to preserve privacy. The general steps of our client-server based privacy aware ML solutions are the following,

- **Data Encoding**. The primary step for privacy preservation in our privacy aware ML solution is to encode the client data in such a way that minimize the information leakage i.e. it makes difficult for any adversary sitting on the server to steal the sensitive information from the data, at the same time the performance of the ML model should not be compromised on the encoded data.

- **ML algorithm redesign**. The next step of our privacy aware ML solution is re-designing the ML algorithm in such a way that it yields comparable results on the encoded space with the original ML algorithm executed on the non-encoded space.

- **Evaluation**. The final step is the evaluation of the privacy aware ML solution. The main purpose of the evaluation is to measure the effectiveness of the ML model and the privacy preservation effectiveness.

A number of solutions has been proposed in the existing literature for privacy preserving data encoding e.g., Bloom filters mechanism [123, 124], distance preserving transformation based mechanism [71, 137], multi-objective learning based mechanism [34], adversarial learning based solution [89], etc. In this thesis we have mainly used distance preserving transformation and multi-objective learning based mechanism to ensure the preservation of data privacy.

Another important requirement for training the ML model using large volume of training samples is the availability of high-end computational resources. To mitigate the computational requirements of high-performance CPUs and GPUs a common practice is to use some publicly available (free or paid) *infrastructure as a service* (IaaS) or *machine learning as a service* (MLaaS) platforms. A client with limited resources can use the MLaaS to train their ML model and provide prediction service through their applications.

To avail the MLaaS the client needs to upload their data to the MLaaS server in raw form or an encoded form which may cause information leakage due to eavesdropping on the communication channel or malicious activity by a malware sitting on the server. The objective of our research is to minimize the information leakage while using the cloud service of MLaaS without compromising the performance of the prediction task. We mainly focus on privacy aware solutions for clustering and classification tasks.

Now, we discuss the main research questions explored in this thesis and conclude the chapter by providing an outline of the thesis.

1.2 Research Questions

The privacy and security concerns of sharing the data over the cloud environment to avail the ML models on MLaaS platforms have been discussed in the previous section. In this section, we briefly present four research questions (RQ-1, RQ-2, RQ-3 and RQ-4) which will be addressed adequately in this thesis. In fact, all our research questions that have been studied here concern the problem of employing machine learning techniques under privacy preservation constraints. Although a number of de-identification or anonymizing methods have been proposed for privacy preservation in the existing literature, most of them are based on adding noise or masking the sensitive information in the data [154, 93]. Encoding of data on an anonymous space is another popular strategy used for privacy preservation of data [137]. The choice of the topic of this thesis has been motivated by the hypothesis that the encoding of data on an anonymous space can minimize the information leakage to enhance the privacy and security.

Locality sensitive hashing (LSH) is a popular method of binary encoding or Hamming space transformation, and according to Johnson-Lindenstrauss (JL) lemma this transformation is distance preserving [167]. Since LSH uses randomized basis vectors to transform the real valued vectors into the Hamming space and without knowing the basis vectors it is computationally difficult to formulate an inverse transformation from Hamming space to the original real space [6], consequently, the notion of privacy preservation comes into account. Thus, in our first research work we employ LSH based Hamming space transformation to preserve the privacy. But the Hamming space is a lossy transformation, so standard K-means could not perform well with the incomplete information on the binary space (considering the binary space as real space). Here, we specially use the projection statistics along the basis vectors of LSH transformation and redesign the unsupervised learning algorithm, in particular K-means algorithm, to improve the efficiency of the learning strategy. Thus our first research question can be stated as,

- **RQ-1:** *How unsupervised learning algorithm can be re-designed under the constraint of privacy preservation to improve the learning effectiveness?*

The second research question, studied in this thesis, explores the effect of weak supervision on the encoding mechanism under privacy preservation constraints. To address this research question the triplet network has been used for realization of the effect of weak supervision. Similar to our studies pertaining to **RQ-1**, here too we have used LSH based transformation of the input data to ensure the privacy preservation and the effectiveness is measured on clustering task. The difference in the approaches dealing with RQ-1 and RQ-2 is the nature of the training procedure, i.e., first one is unsupervised and later one is semi-supervised. Thus the second research question can be stated as:

- **RQ-2:** *How the effectiveness of privacy preserving clustering on discrete metric space can be improved with weak supervision on the encoding transformation?*

In our first and second research questions, we address the problem of privacy preservation in unsupervised and semi-supervised learning scenarios respectively. Now, the third research question dealt here is directed towards exploring whether supervised learning can be used to protect the privacy of the sensitive data. To address this research question we have proposed an informative subspace based multi-objective approach to obtain privacy aware encoding of the data. The main objective of this research is dual in nature, the first objective is to improve the effectiveness of the classification tasks and the second objective is to encode the data in such a way that an adversary fails to predict the sensitive information from the encoded data. Thus we formalized our third research research question as:

- **RQ-3:** *How supervised learning can be used to defend the malicious attempts of stealing sensitive information from data shared on cloud platforms?*

The last research question, studied in this thesis, is about exploring the potentials of applying approximate nearest neighbour (ANN) based indexing and retrieval strategies under privacy preservation constraints to trace the susceptible people who might have been infected by the virus due to their close proximity with people who were recently tested positive towards the infection during a pandemic situation caused by the same virus. To answer this research question, we have developed a laboratory based reproducible environment and conducted experiments on both real and synthetic sample datasets. This last research question dealt in this thesis can be more formally stated as:

- **RQ-4:** *What is the feasibility of applying approximate nearest neighbour (ANN) based indexing and retrieval approaches under privacy preservation constraints to obtain a list of top-k suspected users, who might be infected by an infectious disease, in real time during pandemic?*

1.3 Thesis Contribution

In this thesis, we aim to address the problem of improving the data privacy and security while sharing the data over the cloud environment to avail the Machine Learning model. We now enlist our contributions in this thesis.

- First, we have proposed a novel modified K-means clustering algorithm (privacy preserving K-means) which respects the privacy of the data. A binary encoding i.e. Hamming space transformation of the input data vector has been used to preserve the data privacy. We mainly devise a novel Gaussian mixture model based solution to estimate the new centroid vector on the Hamming space during each iteration. For better estimation of the bit value of each centroid vectors we make available the projection statistics, in form of mean and standard deviation, of the input vectors along a set of random basis vectors.

- Second, we have designed a novel weakly supervised approach of learning the encoding of the input data to improve the clustering effectiveness on Hamming space. Similar to the privacy preserving K-means clustering algorithm (c.f. previous point), here too we have used the binary encoding (Hamming space transformation) of the input data to preserve its privacy. In this work, we empirically demonstrate that an effective reconstruction of the encoded data can be achieved with the help of the projection (along a set of random basis vectors) statistics of the real valued input vectors collected at the time of Hamming space transformation. We also show that this reconstructed data along with a small seed set of non-encoded data can be used to learn a more effective encoding which yields a better clustering results compare to the binary encoding.

- Third, we have proposed a novel informative subspace based multi-objective approach to generate privacy aware encoding of the input data. In this approach we hypothesize that removing or down-weighting the information necessary to determine the sensitive attribute values potentially improves the defence against the malicious attempts of stealing sensitive information from data. In contrast to the existing approach [34, 89], where only the multi-objecting learning has been used, we have applied an informative sub-space selection corresponding to the primary task and multi-objective learning simultaneously to produce a more robust encoding which is more resilient to security threats.

- Fourth, we have investigated the feasibility of applying approximate nearest neighbour algorithms based indexing and retrieval approaches for contact tracing under privacy preservation constraints during epidemics. In this framework we applied the locality sensitive hash function (LSH) followed by a quantization of the projected values of the input vector along a set of randomly chosen basis vectors to preserve the data privacy.

1.4 Thesis Outline

The organization of the thesis is as follows.

- **Chapter 2** presents a comprehensive literature survey of the related works and highlights the difference of our proposed methods with the existing approaches. In particular, we revisit various existing privacy preserving data encoding strategies and various computing mechanisms. Popular techniques like differential privacy, homomorphic encryption, etc. have been reviewed in some details.

- **Chapter** 3 provides brief descriptions of K-means clustering, Hamming space transformation, metric learning, multi-objective learning, etc. The Hamming space transformation has been used in designing solutions of our research questions **RQ-1** and **RQ-2** while the multi-objective learning has been used in dealing with the research question **RQ-3**. Finally, we have discussed the Hierarchical Navigable Small World (HNSW) as well as KD-tree search algorithms which have been used in the proposed Privacy Aware Approximate Nearest Neighbor Search (ANNS) in connection with the research question **RQ-4**.

- **Chapter** 4 provides an overview of the datasets used in the experiments of the subsequent chapters in this thesis. Here, we describe about the synthetic 2D points dataset (viz. Spiral, ΛV and Flame), MNIST-8M synthetic image dataset, ODPtweets real text dataset, which are used in the experiments for clustering task of our research questions **RQ-1**, while MNIST handwritten numeral image dataset and 20-Newsgroups text dataset used to evaluate the clustering efficiency in the research question **RQ-2**. Next, we describe the Skin Cancer MNIST dataset of real images, Morpho-MNIST image dataset (which contains real handwritten numeral image samples of MNIST dataset, and some additional samples generated synthetically by applying morphological erosion on the images of real samples), TrustPilot Dataset containing samples of real texts of US english product review. These three datasets have been used to evaluate the classification task of our research questions **RQ-3**. Finally, we have provided descriptions of the FourSquare - NYC and Tokyo Check-ins dataset and the synthetic trajectory dataset, which have been used to experiment with the ANN-based indexing and retrieval techniques in connection with the research question **RQ-4**. Also, this same chapter introduces the metric used to evaluate the effectiveness of the experiments performed for the work of this thesis.

- **Chapter** 5 proposes an approximate K-means clustering algorithm to address the research question **RQ-1**. The proposed algorithm ensures the privacy preservation of the data using a binary encoding or Hamming space transformation. The clustering memberships of the encoded data are the representatives of the clusters of the original data. The proposed approximate K-means clustering algorithm uses projection statistics, which are collected during Hamming space transformation, for obtaining a better approximation of the centroid over the Hamming space. Details of centroid re-computation steps are provided in the Section 5.2. The cluster assignment process in the algorithm is performed based on the Hamming distance between the binary encoded vectors and the recomputed centroid.

- **Chapter 6** explores a semi-supervised way of generating the privacy aware encoding of the data which address our second research question **RQ-2**. Similar to research question **RQ-1**, here too we have used the Hamming space transformation to preserve data privacy. Also, here we have shown that a set of triplets generated from the output of the clustering algorithm on small amount of non-encoded data can be used to generate clustering aware encoding of the data which yields more improved clustering results on the remaining samples. Finally, we present the experimental results on three standard image datasets namely MNIST, Fashion-MNIST, CIFAR-10 and a text dataset viz. 20-Newsgroups which clearly demonstrates the effectiveness of our proposed method.

- **Chapter 7** addresses our third research question **RQ-3**, where we have used a supervised method to produce the privacy aware encoding of the data. In this chapter, we have presented a method to minimize the leakage of sensitive information from the data using the multi-objective learning approach. We have also shown that the use of an informative subspace based approach along with the multi-objective learning can be more beneficial to obtain privacy aware encoding of the data. Our main hypothesis of this research is that the multi-objective based defence mechanism can be improved by using a scheme for weighting the features. It has been explained in detail in Section 7.3. Finally, we have presented a number of experimental results, conducted on both image (Morpho-

MNIST, Skin Cancer MNIST) and text (Trustpilot) data, to support the claim.

- **Chapter 8** presents the last research question **RQ-4**, which concerns an interesting application of preparing a list of susceptible people who may have come in close contact with persons who have been recently tested positive in respect of the highly infectious virus. We have used the user locations in terms of 4-dimensional Euclidean vector space (3 dimensions for space and 1 for time) and have designed the contact tracing as a search problem in the same vector space. Since the user location data is sensitive in nature, we have applied a quantization transformation on the data to preserve the privacy. KD-tree, an approximate nearest neighbour approach has been used, for the retrieval purpose. Also, the ground-truths for FourSquare Check-ins dataset has been simulated. A synthetic trajectory dataset of user locations has been generated using random walk for their use in the evaluation of the effectiveness of similar approximate approaches.

- **Chapter 9** concludes the thesis by summarizing the research achievements and providing directions for future research. In this chapter, we first revisit each research question and summarize how each one of them has been addressed through the experimental findings described in the corresponding chapters. Finally, we have discussed our ideas for possible enhancement of various methods presented in various chapters of this thesis.

Chapter 2

Related Work

In this chapter we present a comprehensive literature survey of related works. We first revisit various privacy preservation techniques used in data mining. Next, we review some related studies of data clustering and finally we explore the latest advancements in the area of adversarial learning and metric learning.

2.1 Privacy Preservation in Data Mining

It has been quite some time that the concerned group has felt the need for privacy preservation in online processing of data over the internet. The issue is more serious when data are stored in public servers and shared among different groups of people. During the past several years a number of sophisticated techniques have been proposed in the literature for preservation of data privacy. Some of the most popular approaches of data privacy preservation are based on homomorphic encryption, data perturbation, privacy preserving encoding etc. to name a few.

2.1.1 Homomorphic Encryption

Homomorphic data encryption strategies allow its users to perform various computer processing tasks over the data without requiring its security keys. Recent advances in *homomorphic encryption* have made it possible to use as-a-service-based solutions without compromising the confidentiality of sensitive information [116, 98, 145, 142]. Our data privacy preservation approach over cloud computing environment is functionally different from that of the homomorphic encryption, where the objective is to encrypt data in such a way that a certain set of operations (e.g., backpropagation based gradient updates in supervised learning) conducted on the encrypted data is likely to yield approximately similar results in comparison to the results obtained with the data in its original form [87, 119]. In [87], the authors have proposed a homomorphic encryption framework that has been claimed to realize privacy preservation for machine learning training and classification in data ciphertexts environment. The authors of [119] presented a functional encoding scheme. They used its implementation to build privacy preserving neural networks, and tested the same successfully on simple image classification problems. Also, these authors provided an adversarial training technique to improve the privacy by reducing the information leakage. Our workflow, instead, corresponds to the situation when a client needs to communicate with a server without involving data encryption and decryption with key exchanges.

2.1.2 Differential Privacy

Among *data perturbation* techniques, differential privacy (DP) [43] is the most successful privacy preservation technique that produces a strong privacy guarantee before making the data public. The objective of DP is somewhat similar to that of privacy-preserving encoding. DP protects an individual's privacy while publishing information about a database. However, differential privacy does not involve any encoding of the raw data as vectors; instead, it obfuscates parts of relational data so as to mitigate individual data leakage [41]. Various de-identification or anonymizing technologies have been proposed to protect data privacy, which often involve adding noise or masking sensitive information in the released dataset [154, 93]. Some well known DP techniques include adding noise by the Laplacian and the Gaussian mechanisms [39, 43]. While the Laplacian approach is particularly suitable for sparse vectors [42], the Gaussian one finds applications in empirical risk minimization algorithms [9]. The concept of additive noise in differential privacy for relational databases also finds applications in Bayesian risk minimization in general [37], or in Bayesian linear regression [16], in particular. Our proposed methods are different from the concept of DP in the sense that instead of data perturbation we encode the data to preserve privacy.

2.1.3 Privacy-preserving Data Encoding

Privacy preserved data encoding finds applications in record linkage [71, 72], clustering [19, 131], text classification [34], regression analysis [69] etc. For text data, privacy-preserving based encoding is particularly crucial because the inherent characteristics of natural language (e.g., writing style or word usage patterns) often reveal information about the authors, which can be used by adversaries to reveal such sensitive information. As examples, the authors of [118] used online behavior, stylistic choices and language models to predict the age group of blog authors, while those of [113] used Twitter content to predict the occupational class.

It has been shown that a multi-objective approach, where an adversarial classifier model is trained simultaneously with the primary task classifier, is useful to obtain a privacy preserving encoding of the data [34]. Other work on data encoding has applied distance preserving binary encoding of data instances [71, 72, 19].

Multi-objective approach. A number of recent studies has proposed the dual objective of privacy preservation (minimizing leakage of sensitive information) and model preservation (maximizing the performance of an algorithm on the encoded data), e.g., applying a 'multi-detasking' model to train an adversarial classifier simultaneously with the primary downstream text classifier, where during training, the primary classifier updates its parameters to confuse the attacker model [34]. The study reported in [92] developed a distributed framework for privacy preserving multi-task learning protocol by applying encryption mechanisms. The authors of [89] explored an adversarial learning approach that learns unbiased representations of text with respect to specific sensitive attributes. Somewhat different from the findings of [34], the authors of [45] showed that despite adversarial training methods being generally effective in reducing the amount of implicit sensitive information, in some cases, however, a substantial amount of sensitive information still persists and can be extracted from the encoded representations.

Although our proposed privacy preserving classification model falls into the general class of multi-objective approaches, such as those of [34] and [127], our proposed method is more general in the sense that we leverage the candidate subspaces that are most informative of the primary task. Since parts of these subspaces are less likely to be comprised of the sensitive information in data, our method seeks to address some of the concerns pointed out in [45], i.e. removal of sensitive attributes (e.g. demographics) from data instances can still lead to an adversary predicting this missing information. Our subspace-based approach is explicitly directed towards mitigating this problem in the sense that the privacy-aware encoding process puts more emphasis only on those components of the data that are more useful for the primary task, while suppressing the residual space that contains most of the information on the sensitive attributes. Standard approaches of model-agnostic instance-wise informative feature selection for classification include those of employing linear regression to learn a simplified decision boundary by sampling points around a data instance [97], applying a Gumbel distribution to estimate instance-wise feature importance [30] etc. The authors of [52] reiterate the importance of feature selection for supervised learning tasks, whereas those of [90] and [171] explore feature selection for the case of unsupervised learning. In the context of our work, we use the idea of exploring informative candidate subspaces with a parameterized approach, as first proposed in [30]. An explicit use of feature importance also provides an interpretable way of preserving data privacy.

Distance preserving binary encoding approach. A popular technique to privacy preserving data encoding is distance preserving Hamming space transformation due to its computational and memory efficiency. Among the various binary transformation e.g. Locality Sensitive Hashing (LSH) [56], Spectral Hashing (SH) [160], Binary Reconstructive Embeddings (BRE) [80], Shift Invariant Kernel based Hashing (SIKH) [114], etc., LSH is popularly used technique, where similar samples are mapped into the same bucket with high probability. In other word, using LSH the original metric (e.g. Euclidean distance, cosine similarity) is well preserved in the Hamming space with increasing code length. The LSH consists of projection along the randomly chosen hyperplane and thresholds. The Super-Bit LSH proposed in [68] improves the likelihood of semantic hashing (similar signatures corresponding to similar points and dissimilar signatures otherwise) by applying orthogonalization of the randomly chosen basis vectors with the help of Gram-Schmidt algorithm. Since our proposed privacy preserving semi-supervised and unsupervised clustering methods uses the Hamming space transformation to preserve the privacy, among the other alternatives we employ the Super-Bit LSH as the transformation function.

2.2 Privacy Preserving Clustering

In recent years, the research of privacy-preserving computing has received considerable attention, such as linear regression [69], K-means [54] etc. The work in [69] learns a transformation function to simultaneously maximizing the likelihood of predicting missing values from the data and also minimizing a linear regression loss. Two major differences of our privacy preserving clustering method with respect to [69] are that, firstly, we focus on a different objective, namely that of *clustering*, which in contrast to the objective of linear regression in [69], is unsupervised in nature, and secondly, the transformation function in our case is a binary one instead of a low rank approximation of [69], thus ensuring much faster execution.

In contrast to our client-server setting of K-means computation, the authors in [54] address the distributed computing case where the K-means computation is securely distributed over computing resources before employing secure key exchange protocols for computing the centroids and the closest cluster centres. Researchers in [109] proposed an attribute generalization based algorithm to abstract out specific instances of values of attributes, e.g. replacing attribute values such as 'dancers' and 'writers' with the more general value 'artists'.

Similar to our approach of binary encoding the data with additional information about the averages and the variances of values projected along basis vectors, the work in [40] shares additional information of the form $f(x) = g(x_i)$, where x_i denotes the i^{th} row of a database and g maps database rows to $[0, 1]$.

K-means on Hamming Space. Since our proposed privacy preservation based K-means is based on binary encoding of data, we now review some existing K-means clustering variations that work with binary data. For example, the work in [74, 131] represented data vectors as binary codes to perform clustering. While the study in [74] defined a cluster centroid as the component-wise median of constituent vectors of a cluster, the authors of [126] obtained sparse cluster centers by applying $L1$-ball projection on each cluster center during each mini-batch iteration of K-means, which contributed to reduction in computational cost.

The idea of PQK-means involves representing input real-valued vectors as short codes by applying product quantization [61] and then clustering them by making use of hashing on the PQ codes during the cluster assignment step and sparse voting during updating the centroids [48]. The main limitation of PQ codes is that it has to rely on fixed subspaces of the data. In contrast, the JL transformation [161, 68] used to encode the data vectors in our method is able to preserve more information about the data by taking projections along orthogonal basis vectors. Similar to our method, the study in [131] uses random basis vectors to encode the input data in binary. However, during intermediate steps, the algorithm makes use of the original data vectors to modify the basis vectors, which leads to violating the privacy preservation constraint.

2.3 Deep Clustering

Although the initial works of clustering were based on traditional machine learning techniques, researchers have also employed deep neural methods for this purpose. In an earlier study, the authors of [99] presented a neural network based method, namely the MacLeod algorithm, for document clustering. The MacLeod algorithm, broadly speaking, first classifies if a document belongs to a cluster, and then if so, adjusts the network parameters to make the document vector ‘move closer’ to the cluster centroid. The authors reported that the algorithm yields comparable results to hierarchic (sequential) clustering algorithms.

Several approaches have been proposed for simultaneously performing clustering as well as subspace selection. A kernelized K-means algorithm was proposed in

[166] that involves a supervised subspace selection using linear discriminant analysis (LDA). The LDA method enables a class-aware dimensionality reduction and has been shown to improve clustering effectiveness. The experimental results on some benchmark datasets shown that the kernelized K-means yields comparable performance with standard clustering algorithms such as K-means.

Likewise, an unsupervised feature selection based on trace ratio formulation is proposed in [152] to simultaneously execute the subspace selection and clustering. These methods usually apply a shallow linear embedding function.

Existing research on deep representation learning has also investigated obtaining an embedding of a data space with an objective of assigning cluster labels [55, 163, 62, 168] with specific applications in community detection on networks using non-linear embedding functions [25]. While some of these semi-supervised techniques use statistical measures, such as the KL divergence [55, 163], the others are based on triplet networks [62, 168].

Our work differs from the aforementioned semi-supervised work on deep metric learning in two significant ways. First, in contrast to semi-supervised learning, we do not employ a subset of the data labels, and second, the input on which the metric is learned represents an encoded space in our workflow of privacy preservation based clustering.

2.4 Adversarial Learning

An adversarial attack broadly refers to the methods of generating samples (often called adversarial examples) that are indistinguishable from samples drawn from the true data distribution with an objective to 'fool' a classifier [58]. These attacks typically use first order gradient information, such as FGSM [58], I-FGSM [81], MI-FGSM [38], Ada-FGSM [132] etc. Successful demonstrations of black-box adversarial perturbations attacks leading to degrading the effectiveness of classifiers were demonstrated in [110] and [86]. Defence mechanisms against such adversarial attacks include those of using regularized FGSM [141], and defensive distillation [111].

Different from adversarial learning, in our supervised classification model, we rather employ a multi-objective encoding, the purpose of which is to ensure that it potentially would be difficult for an adversary to use a pre-trained system (on similar data) to effectively predict the values of sensitive attributes (e.g., age, gender etc.) from the encoded data.

2.5 Metric Learning and Triplet Networks

Learning a data-driven similarity metric is an active area of research [170, 53, 122], and has applications in nearest neighbour classification [159], clustering [62, 168, 164], face recognition [32], person re-identification [60, 168], ad-hoc document retrieval [59], image-text retrieval [88], etc.

In [155], the authors proposed a graph convolution based clustering of multi-view data. More precisely, they first conducted clustering on the vector representations of the graph convolution network (GCN) latent layers for each view, and then they imposed an additional constraint seeking to optimize the clustering results iteratively. An additional constraint in their work assumed a joint representation of each view of the data in the same semantic space.

While the early metric learning techniques, such as the large margin nearest neighbor (LMNN), involve shallow approaches such as employing an SVM (hinge loss), recent approaches have applied neural learning to rank approaches [23] for the purpose of metric learning [125, 60, 62, 168]. The loss function, typically useful to train such networks, makes use of triplets, comprised of a pivot point along with a positive and a negative example. The objective of a triplet network is to learn the parameters of a distance metric that minimizes the distance of the positive sample from the pivot while simultaneously maximizing the distance of the negative sample from the pivot.

2.6 Proximity Tracing

Recent advancements in homomorphic encryption has made it possible to perform computation on encrypted data at the server end without compromising the sensitivity of data [115]. Under such a privacy-preserving workflow, only the encrypted data is shared with the server and all computation is restricted to use only the encoded data [116, 4, 18, 13, 134]. Among existing work, [116] used homomorphic encryption on mobile IOT systems to preserve privacy. Other applications of homomorphic encryption based contact tracing include those of [4], which uses Bluetooth signal, and [13], which uses WiFi identifiers etc.

Government of different countries also took initiative to develop secure contact tracing application e.g. TraceTogether[1], COVIDSafe[2], Aarogya Setu[3] etc. by Singaporean, Australian and Indian governments respectively. Some companies such as Google and Apple also released APIs[4] to support contact tracing and many countries agreed to using these apps as a part of their policies.

2.7 Approximate Nearest Neighbors Search

Existing studies on nearest neighbors (NN) search attempt to find the closest K points to a given query point from a dataset. The KD-tree algorithm, proposed in [14], is one of the most popular *exact* nearest neighbors (NN) searching algorithm which can be converted to ANN by applying an upper bound on the number points to be examined. Although it yields good results in low dimensional spaces, its effectiveness in terms of computation time and memory usage, tends to decrease for high dimensional spaces [29]. Since exact nearest neighbor (NN) search algorithms (e.g. the classic KD-tree) being computationally expensive, are rather intractable for large collections of embedded data in high dimensional spaces. This has motivated research towards towards research on approximate NN retrieval.

ANN search finds applications in content-based image retrieval. With the advent of deep learning based methods which represent image and text data in a joint embedding space of reals [50], finding nearest neighbors in the data can be useful for various applications, such as image captioning [73], imagification of documents [2] etc.

[1] https://www.tracetogether.gov.sg/
[2] https://www.covidsafe.gov.au/
[3] https://www.aarogyasetu.gov.in/
[4] https://covid19.apple.com/contacttracing

Generally speaking, existing ANN approaches can broadly be divided into the following categories. Firstly, some approaches are memory-based relying on efficient data structures to compute only a limited number of exact distances [33]. Variations of the KD-tree data-structure to support ANN e.g. classic KD-tree [133], Best Bin First (BBF) search [8], balanced box-decomposition (BBD) tree [12] also fall in this category.

The second category of approaches are hash-based which aim to design effective hash functions to preserve the spatial proximity of points, i.e. map close points to the identical hash values. Examples include Locality sensitive hashing (LSH) [6], LSH Forest [11] etc. LSH [6] is the most popular hashing-based ANN search method which uses a number of different distance preserving (also called semantic) hash functions.

The third category of approaches map data points to compact binary codes to reduce in-memory space and achieve fast exhaustive search in Hamming space [57]. Product quantization (PQ) [61] is a specific type of non-binary discrete encoding method used for either exhaustive search or non-exhaustive search with the help of inverted indexing.

The fourth category of approaches is based on metric inversion (MI), i.e. those that rely on pre-computing distances from a set of reference points (different from the data points). These distances are stored in the postings list corresponding to each reference term [5]. Among more successful approaches that allow provision for an inverted index based secondary storage organization (with query driven dynamic loading of content in the primary memory) are the graph-based approaches - NSW and HNSW. Navigable Small World (NSW) [22] is a graph with logarithmic or poly-logarithmic scaling of greedy graph routing [22]. NSW-based ANN search was further improved in [100] with a controlled hierarchy based approach, known as the Hierarchical NSW (HNSW).

2.8 Trajectory Search

Recent advancement in GPS technology has led to everyday recording and storing large amounts of trajectory data of moving objects. This high volume trajectory data can can potentially be leveraged for trip recommendation [129], travel time and path optimization [157], identifying driver expertise [136] etc.

There also exists a number of recent studies about trajectory search given a particular query location [143], region of interest [128] or traveler's preference [130] and activity [156].

Our use-case is different from the above thread of work in the sense that for a given trajectory, instead of searching closest k complete trajectory, our task is to retrieve k users whose spatial-temporal data values (considering both GPS location and time) are close to that trajectory.

2.9 Conclusions

In this chapter, we have discussed several existing studies related to the research questions introduced in this thesis and published in recent literature. In the next chapter, we will discuss some of the necessary preliminary concepts related to the research questions described in previous Chapter while in the succeeding chapters we will discuss these research questions in more details and propose solutions for them.

Chapter 3

Preliminaries

This chapter presents briefly a few terminologies which have been used in the works of the subsequent chapters of this thesis. It start with an overview of Euclidean and Hamming space, followed by encoding of real-valued vectors from the Euclidean space to the Hamming space. We then present K-means clustering which is perhaps the most popular one among similar techniques and discuss about multi-objective learning, mutual information between two random variables and metric learning, which are used to address the second and third research questions. Also, it includes brief introductions of Triplet Networks, feature subspace selection, KD-Tree and Hierarchical navigable small world (HNSW).

3.1 Euclidean and Hamming Spaces

A vector space $\mathbb{V}$ is represented by a set of vectors $\mathcal{V}$, with each component belonging to a specific domain (e.g. real numbers) and a distance metric $\mathcal{D}$, which takes two vectors as input and outputs a non-negative real number. More formally,

$$\mathbb{V} : (\mathcal{V}, \mathcal{D}); \;\; \mathcal{D} : (\mathbf{x}, \mathbf{y}) \mapsto \mathbb{R}, \; \mathbf{x}, \mathbf{y} \in \mathcal{V}. \tag{3.1}$$

The two vector spaces that are relevant in the context of our problem of privacy preserving clustering are

(i) $\mathbb{R}^d$: a d dimensional real vector space with L_2 (Euclidean) distance metric, and

(ii) $\mathbb{H}^m$: an m dimensional Hamming space of vectors with binary (0/1) components with L_1 distance metric, commonly known as the Hamming distance ($\mathcal{D}_H$).

As notations, let $V = \{\mathbf{v} : \mathbf{v} \in \mathbb{R}^d\}$ be a set of points in a d dimensional real-valued Cartesian space. In our description of the privacy aware clustering approach, we make the general assumption that the data instances are real-valued vectors. This allows provision for addressing both text and images as inputs, both of which can be converted to dense vectors with the application of standard methods, e.g., text can be converted to a dense vector with the application of LSTMs [96], or an image can be converted to a vector by the application of a variational autoencoder [76].

3.2 Binary Encoding of Data Instances

Explicitly sharing the set of real-valued vectors to an MLaaS compromises privacy because of the potential presence of sensitive information within the data. A standard approach of sharing the data with a server is thus to first encode the data by transforming points from one space to another. A transformation is usually meaningful if it is *distance preserving*, i.e., two nearby points continue to remain in close proximity post transformation. The research challenge is thus to make the clustering algorithm on the server side work effectively with this encoded data. Formally speaking, we transform a d dimensional vector $\mathbf{v} \in \mathbb{R}^d$ in Euclidean space, into a binary vector $\mathbf{h}$ in the Hamming space of dimension m, ($\mathbf{h} \in \{0, 1\}^m$), which we interchangeably denote as $\mathbb{H}^m$. Let ϕ be the transformation function, i.e.,

$$\phi : \mathbf{w} \in \mathbb{R}^d \mapsto \mathbf{x} \in \mathbb{H}^m \tag{3.2}$$

Among various alternatives of the selection of the function ϕ, e.g. with random projections [161] or with hash-based methods [6], we specifically employ the Super-Bit locality sensitive hashing (Super-Bit LSH) algorithm [68].

3.2.1 Super-Bit LSH

The Super-Bit LSH algorithm first selects a set of m random basis vectors $\mathcal{B}$ and then orthonormalizes them by applying the Gram-Schmidt algorithm [68]. Next, it involves taking the projection along each basis vector $\mathbf{b}_i \in \mathcal{B}$. In other words, for a data instance vector $\mathbf{v} \in \mathbb{R}^d$, the i^{th} bit of the encoded vector $\mathbf{h} = \phi(\mathbf{v})$ in the transformed Hamming space $\mathbb{H}^m$ is computed as

$$\mathbf{h}_i = \text{sgn}(\mathbf{v} \cdot \mathbf{b}_i), \;\; i = 1, \ldots, m, \tag{3.3}$$

where $\mathbf{b}_i$ represents the i^{th} basis vector and $\text{sgn}(\cdot)$ is the sign function that returns 0 for a negative value of the parameter, and 1 otherwise.

3.2.2 Characteristics of LSH-based Binary Encoding

A binary encoding function, in general, is distance preserving. This can be realized with the help of the Johnson-Lindenstrauss (JL) lemma which shows that LSH-based transformations (including the Super-Bit algorithm, in particular) is distance preserving [167].

Another desirable characteristic of an LSH-based binary encoding is that such a transformation leads to privacy of the data being preserved, where the notion of privacy preserving computing, which we use in this thesis, relies on the observation in [6] that without knowing the set of basis vectors, it is computationally difficult to find an inverse function of the JL transformation transformation ϕ, that we use (Equation 3.2) to encode the vectors before sending them to the server.

At this point, we mention that in contrast to the standard notion of *differential privacy*, which applies for relational data comprising a set of attribute-value pairs, in our work, we consider privacy preservation (specifically during K-means computation) of real-valued data only. This means that instead of enforcing differential privacy, all we need to ensure in the context of our problem is that it should be difficult for an adversary to compute the true data vectors from the encoded vectors sent to a server for K-means computation.

3.3 K-means Clustering

Clustering is one of the classic problems in machine learning and pattern recognition. The objective of the clustering is to partition the dataset in such a way that the similar data points being assigned to the same group i.e. in other words, the algorithm assign a label to each points of the dataset such that it assign the same label for the similar data points and different labels for the dis-similar points. According to a recent survey of data mining techniques the K-means algorithm “is by far the most popular clustering algorithm used in scientific and industrial applications” [15]. The standard K-means algorithm is the Lloyd’s algorithm [94]. It is a simple and fast algorithm that seeks to minimize the average squared distance between points in the same cluster. It consists of following steps:

Step 1. Randomly choose K points $\{c_1, c_2, \ldots, c_k\}$ as initial centers.

Step 2. Partitions the input data points into K clusters $C_i; i = 1, \ldots, K$ by assigning the points which are closer to c_i than they are to c_j for all $j \neq i$.

Step 3. For each cluster C_i recompute the center c_i as $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

Step 4. Repeat Steps 2 and 3 until the centers no longer changes.

It is standard practice to choose the initial centers from the input data points and if there are ties in Step 2 that may be broken arbitrarily.

In general, the K-means algorithm is designed to work well in the real Euclidean space i.e. the Euclidean distance metric among the centers and points is used to assign the cluster label of the points and the new centroid produced by the centroid re-computation step being belongs to the real space. In our research question RQ-2 the working vector space are real Euclidean space and Hamming space but the final working space is real Euclidean space so we perform the K-means algorithm on real Euclidean space. But in case of the first research question RQ-1 the working space is Hamming space, so we need to re-design the K-means algorithm to apply it on Hamming space. Specially in this case we have used the Hamming distance to assign the class label and proposed a novel method to recompute the centroid such that the new centroid remain a points in the Hamming space.

3.4 Metric Learning and Triplet Loss

Measuring the distance or similarity between data is of utmost importance in machine learning, pattern recognition and data mining, but it is usually difficult to create such good metrics for specific problems. So, metric learning has attracted significant attention. The main objective of metric learning is to find a function which maps points from the data space into a embedding space such that the simple distance, e.g. the Euclidean distance, in the embedding space approximates the "semantic" distance in the data space. The most popular loss function which is widely used in existing metric learning methods is Triplet loss [153, 125, 31], which try to capture the relative similarity among the data points instead of the absolute similarity. To address the second research question RQ-2 in this thesis we have proposed to use weakly supervised metric learning on Hamming space using Triplet loss to produce real embedding of data vectors, which yields a better clustering results under privacy preservation constrain.

3.5 Multi-objective Learning

The Multi-objective learning, often known as Multi Task Learning (MTL), is a learning paradigm where a machine learning model learn multiple related tasks simultaneously and it leverage information from one task to better train the other tasks. More formally, let $\{T_i\}_{i=1}^{t}$ be t learning tasks where all or some of them are related. The goal of Multi-objective learning is to learn all t task simultaneously to improve the model for each learning task T_i using the information achieved from some or all of the remaining tasks. The Multi-objective learning can be realized with a neural architecture, schematically represented in the Figure 3.1. Concretely speaking, a linear transformation is employed to transform each data vector to a shared abstract representation followed by a set of linear transformation specific to each task.

In many existing literature of computer vision and Natural Language Processing (NLP) with different machine learning task it is found that Multi-objective Learning approach outperform their single task counterpart. The authors of [34] have shown

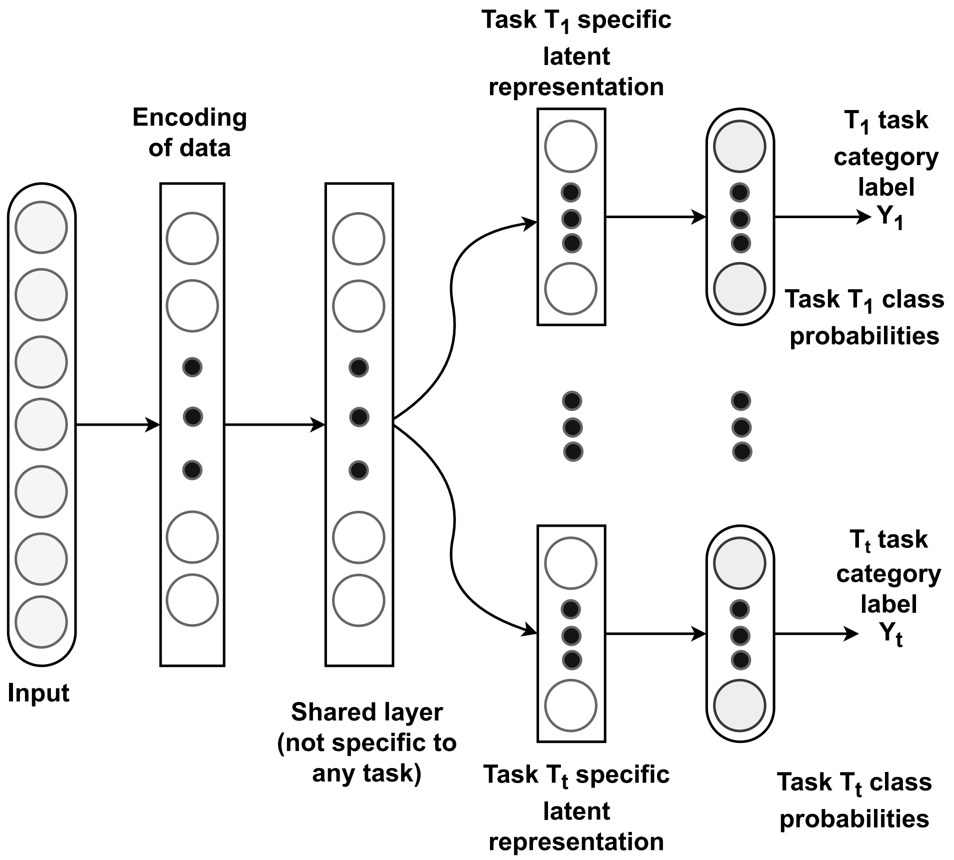

Task T_1 specific latent representation
Encoding of data
T_1 task category label Y_1
Task T_1 class probabilities
T_t task category label Y_t
Input
Shared layer (not specific to any task)
Task T_t specific latent representation
Task T_t class probabilities

Figure 3.1: Schematic diagram of Multi-objective learning

that MTL is potentially beneficial to produce a privacy aware representation of text. It is also found that the performance of the learning task can be improve further by combining MTL with other learning paradigms including unsupervised learning, semi-supervised learning, reinforcement learning, multi-view learning, active learning and graphical models etc. In our present workflow of privacy preserving supervised classification we have combined the Multi-objective learning with informative subspace encoding to improve the defence mechanism against information stealing attacks.

3.6 Feature Subspace Selection and Mutual Information

A basic property of important feature of a dataset is that it carries useful information about all the classes of the dataset. Feature subset selection is an important step of many machine learning model. It often helps to reduce computational overhead, reduce the effect of curse of dimensionality and improves the accuracy of the downstream task.

Mutual information is a widely used criteria for feature subset selection [10], where the selection of feature is performed based on the mutual information between the response variable and the selected features which is to be maximizes [112]. The mutual information $I(X, Y)$ between two random vectors X and Y is a measure of dependence between them. Informally speaking, it corresponds to how much information about one random variable can be obtained by observing the other random variable. It can be measured by the Kullback-Leibler divergence of the product of marginal distributions of X and Y from the joint distribution of X and Y, more formally,

$$I(X, Y) = \mathbb{E}_{X,Y}\left[\log \frac{P_{X,Y}(X, Y)}{P_X(X)P_Y(Y)}\right] \tag{3.4}$$

where $P_{X,Y}$ and P_X, P_Y are the joint and marginal probability densities if X, Y are continuous random variable, or the joint and marginal probability mass functions if they are discrete and $\mathbb{E}$ represents the expectation of a random variable. The expectation is taken with respect to the joint distribution of X and Y.

3.7 Nearest Neighbour Search Algorithms

Here, we provide brief descriptions of Hierarchical Navigable Small World (HNSW) and KD-tree search algorithms which are used in the proposed Privacy Aware Approximate Nearest Neighbor Search (ANNS) strategy presented in Chapter 8.

3.7.1 KD-Tree

KD-tree is a multi-level space partitioning binary search tree data structure, where K is the dimensionality of the search space. Each node in the tree consists of K keys (which comprise the data vector) and two pointers which points to the left sub-tree and right sub-trees. The general idea in KD-tree is to partition a given collection of points by hyperplanes perpendicular to the axes. Associated with each node there is an integer j $(0 \leq j < K)$ called the *discriminator*, the role of which is to determine the direction (left or right) of a data point with respect to the splitting hyperplane (the hyperplane perpendicular to the axis of the j^{th} dimension). The root node has the discriminator value 0. Insertion and searching in a KD-tree take place by recursively traversing the tree and determining the discriminator values at each level by computing the median of the values corresponding to the j-th dimension.

3.7.2 Heirarchical NSW

In general, proximity graph based methods constructs an index by preserving the links to closest neighbours for each individual data point. The basic greedy search algorithm on this proximity graph is very expensive due to the curse of dimensionality and it generally yields relatively poor effectiveness on data with well-separable clusters [100]. To address this limitation, Navigable small world (NSW) graph based algorithm was proposed in [100] for solving the approximate nearest neighbour search problem.

The NSW graph, say $G(V, E)$, is a network with logarithmic or poly-logarithmic scalability of the greedy search algorithm [77], where there is a one-one mapping between the vertex set V of G and the elements of the input dataset $X \subset \mathbb{R}^d$, the set of edges E representing the link among the elements being determined by the following construction algorithm. The edge construction algorithm repeatedly connects a randomly selected node (a data point) with its nearest neighbor. More formally, $(u, v) \in E$ if $\mathbf{x}_u \in N(\mathbf{x}_v)$ or $\mathbf{x}_v \in N(\mathbf{x}_u)$, where $N(\mathbf{x})$ represents the neighbourhood of a point $\mathbf{x}$ in X.

An improved version of the NSW algorithm is the Hierarchical NSW (HNSW) algorithm proposed in [101]. The key idea of index construction and search strategy in HNSW is to extend the graph structure of NSW into a hierarchy of a multi-layered structure, having the links separated by their characteristic distance scales. The HNSW graph is constructed by consecutively inserting a node for each data point, where for each inserted node an integer $l = \lfloor -\ln(\mathcal{U}(0,1))m_L \rfloor$ is chosen to determine the maximum level of the element, where $\mathcal{U}(0,1)$ is the standard uniform distribution and m_L is a normalization factor for level generation.

The insertion procedure has two phases. In the first phase a greedy algorithm starts from the top layer to find e closest neighbours of the inserted element (e is a parameter to control the search quality, its value in the first phase being set to 1). In the second phase, search continues to the lower layers considering the closest neighbours found in first phase as entry points, and the process is repeated. The HNSW ANN search procedure is identical to the insertion algorithm for an element with layer $l = 0$. The search result constitutes the set of closest neighbors found at the bottom-most layer of the underlying structure.

3.8 Conclusions

In this chapter, we have presented certain preliminaries which are essetial to address the research questions introduced in Chapter 1. We have first described Euclidean and Hamming spaces and it has been followed by a brief description of an encoding function, namely Super-Bit LSH, which transforms an input vector from Euclidean space to Hamming space. Next, we have discussed K-means clustering algorithm, which is the target clustering algorithm of our research questions RQ-1 and RQ-2. Also, we have presented an overview of multi-objective learning, triplet network, feature subspace selection and the idea of mutual information between two random variables, which are the key topics necessary to address the research questions RQ-2 and RQ-3. Finally, we have discussed the KD-Tree and Hierarchical navigable small world (HNSW) technologies which have been used in the Privacy Aware Approximate Nearest Neighbor Search algorithm proposed in Chapter 8 to address our last research question RQ-4. Chapter 4, the next chapter, provides brief discussions of the datasets as well as various metrics used in various experiments to show performance of the proposed solutions of the research questions presented in this thesis.

Chapter 4

Datasets and Evaluation Metrics

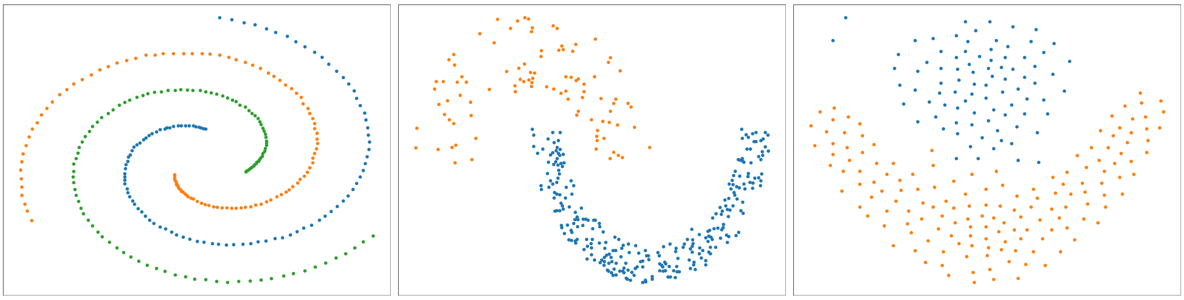
This chapter provides a brief description of the datasets used in the experiments conducted for the present research and the metrics used for evaluation of the results of these experiments. We start with the descriptions of both synthetic and real datasets used in the clustering task. Then we describe the datasets used in the classification task and approximate nearest neighbour search. Finally, we present the evaluation metrics used to evaluate the results of clustering and classification tasks.

4.1 Datasets for Clustering Task

We have conducted experiments for the clustering task on three synthetic two dimensional points datasets containing samples from varying number of clusters. These datasets include ‘Spiral’ [27] containing three clusters, ‘ΛV’ [66] containing two clusters and ‘Flame’ [51] containing two clusters. In addition to using these synthetic two dimensional points datasets, we have also tested our clustering approach on another six datasets of samples taken from the real-world. These are ‘MNIST’ [84], ‘MNIST-8M’ [95], ‘Fashion-MNIST’ [162], ‘CIFAR-10’ [79], ‘ODPtweets’ [7] and ‘20-Newsgroups’ [82], where samples of first three datasets are gray-scale images, fourth dataset is a colour image dataset and the last two datasets consist of text samples. Futher details of these datasets are provided below and Table 4.1 summarizes these details.

4.1.1 Synthetic 2D Points Datasets

Since at each step of our proposed privacy preserving approximate K-means clustering algorithm in the solution of our first research question **RQ-1**, we require to estimate the centroids from incomplete (encoded) information, it is useful to visually compare the estimated centroids at each iteration of the PPK-means algorithm with the true centroids obtained with standard K-means (to be discussed in Section 5.5). For this purpose, we conduct experiments on a number of benchmark datasets in 2 dimensions[1] [27, 66, 51]. Figure 4.1 plots the three datasets used in our experiments of unsupervised clustering algorithm (to be discussed in Chapter 5). The datasets exhibit a range of diversity in the number of visually perceived clusters, the convexity of these clusters and the connectivity between them, e.g., the dataset 'Spiral' (Figure 4.1a) represents 3 disconnected blocks of thin spirals, whereas the dataset in Figure 4.1b comprises two thick 'V' like shapes (one of them inverted), (the reason why we call it 'ΛV'). The dataset 'Flame' (Figure 4.1c) represents two clusters, one of them being convex (top).



(a) Spiral [46] (b) ΛV [47] (c) Flame [91]

Figure 4.1: Visualization of the ground-truth clusters of the two dimensional datasets used in our experiments.

4.1.2 Real Image Datasets

MNIST :- The MNIST[2] [84] dataset consists of 70K images of hand written digits, each represented as a 28×28 two dimensional real valued vector. The number of components (or ground-truth clusters) of this dataset is 10, each corresponding to one of the

[1] The Synthetic 2D datasets are publicly available at http://cs.uef.fi/sipu/datasets/

[2] The MNIST dataset is publicly available at http://yann.lecun.com/exdb/mnist/

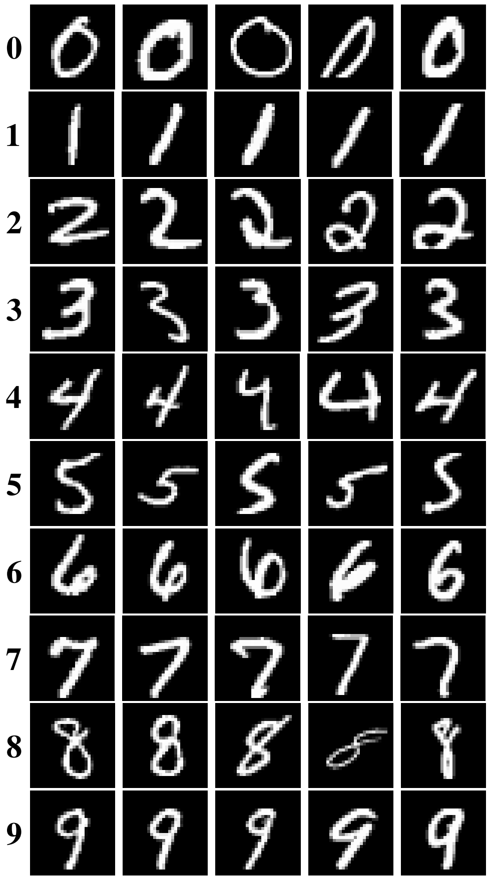


Figure 4.2: Some sample images from MNIST dataset. First printed numeric column is the ground-truth label of the samples of each row.

digits, i.e. $\{0,...,9\}$. Some sample images of each ground-truth cluster of the MNIST dataset are presented in the Figure 4.2.

Fashion-MNIST :- The Fashion-MNIST[3] (referred to as F-MNIST from hereon) [162] dataset is a gray scale image dataset of Zalando's fashion products, which contains 70K images of 10 types of clothing, such as shoes, t-shirts, dresses, and more. Each image is represented as a 28×28 two dimensional real valued vector. Some sample images from each classes of the F-MNIST dataset is presented in the Figure 4.3.

CIFAR-10 :- The CIFAR-10[4] [79] dataset consists of 60K colour tiny images of 10 classes such as airplane, automobile, bird, and more. Each image is a 32×32 colour image. Some sample images from each classes of CIFAR-10 dataset are shown in Figure 4.4.

[3] The Fashion-MNIST dataset is publicly available at https://github.com/zalandoresearch/fashion-mnist

[4] The CIFAR-10 dataset is publicly available at https://www.cs.toronto.edu/~kriz/cifar.html

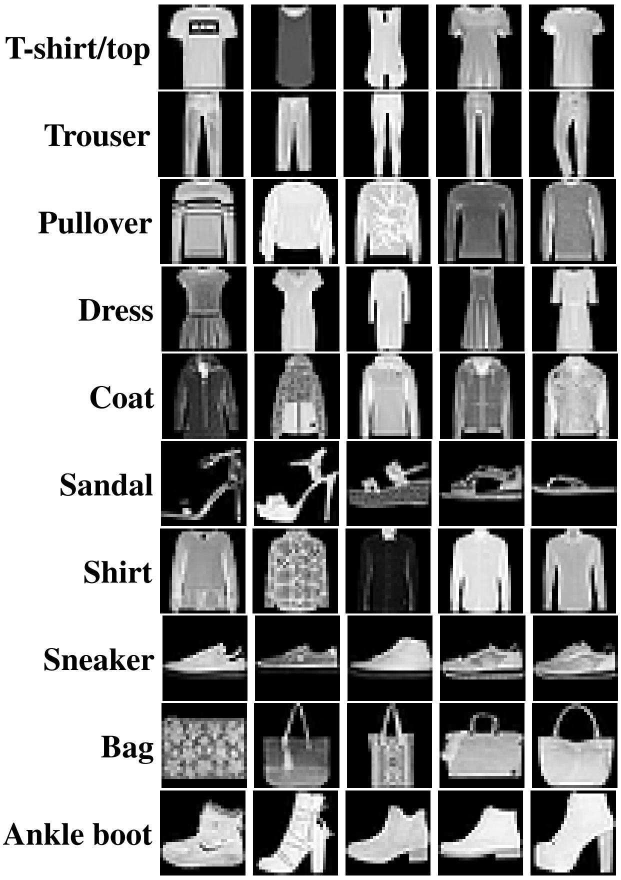


Figure 4.3: Some sample images from Fashion-MNIST dataset. Samples of each row belongs to same class and first column represents the 10 different class names of the dataset.

Since our generic privacy preserving framework accepts dense vectors as input data instances, we converted each image of the MNIST, F-MNIST and CIFAR-10 datasets into vectors of dimension 128 by the application of a VAE, since VAE-based transformation has been shown to preserve the spatial properties of images [76]. We have used these three real image datasets (MNIST, F-MNIST and CIFAR-10) in the experiments of semi-supervised learning framework to show its effectiveness (refer to the Chapter 6).

4.1.3 Synthetic Image Datasets

MNIST-8M :- The 'MNIST-8M' [95] dataset is comprised of 8.1M hand-written digits, each being a 784 dimensional feature vector (a gray-scale image with 28×28 pixels). The MNIST-8M is an extension of the original MNIST dataset of $70K$ images by generating random deformations to the original MNIST images. Consequently, similar to the MNIST dataset the number of ground-truth clusters of this dataset also be 10.

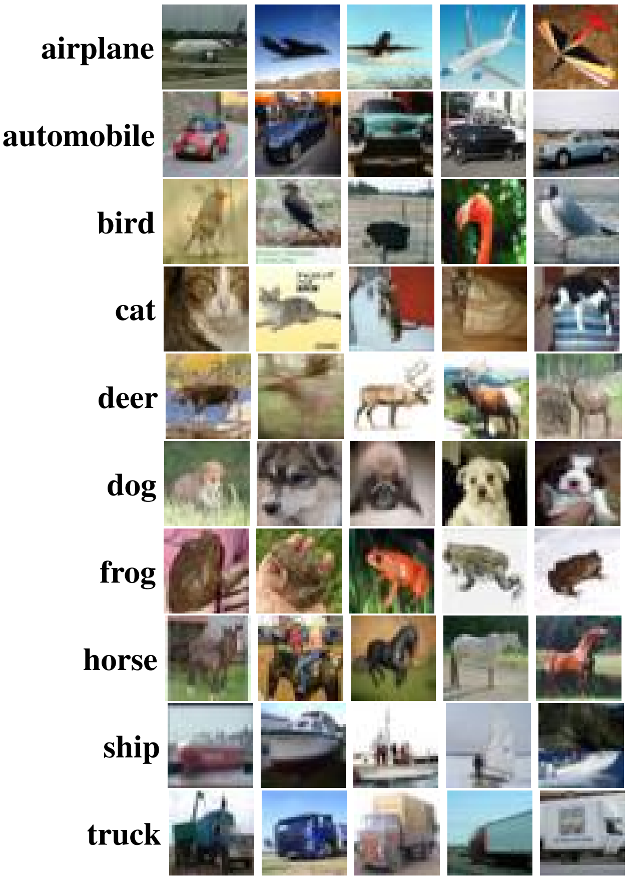


Figure 4.4: Some sample images from CIFAR-10 dataset are shown. Each row contains 5 random samples from same class and the names of of respective classes are shown in the left hand column.

4.1.4 Real Text Datasets

ODPtweets :- Additionally, to evaluate our approaches on text data, we use the ODPtweets dataset[5] consisting of 25M tweets. Each tweet is labeled with the 'Open Directory Project' (ODP) category of the URL of the page which the tweet points to, total number of categories being 34185. On careful observation of the dataset, we found that there is a large number of ODP categories (specifically, 33770) with small number of candidates (specifically, < 10), and that a number of classes (specifically, 12) have an excessively large number of tweets (specifically, 100K). For our experiments, we removed these head and tail categories, which resulted in a total of over 2.1M million tweets distributed among 403 ODP categories. These ODP categories were considered as the ground-truth cluster of the dataset.

20-Newsgroups (20NG) :- In addition to the ODPtweets dataset, we also conduct

[5] The ODPtweets dataset is publicly available at http://www.zubiaga.org/datasets/odptweets/

our experiments on another standard text dataset, namely the 20-Newsgroups (referred to as 20NG from hereon) dataset, which consists of around 18K newsgroup posts on 20 different topics. The number of ground-truth clusters for the 20NG dataset is 20 (reflecting each topic).

In order to obtain feature representations of each tweet and news document, we trained word embedding employing 'skipgram' model of 'word2vec'[6] (with default parameters) over the tweet collection and news document collection respectively using 200 dimensions to represent each word. A dense vector representation of each tweet is then obtained by summing the word-embedded vectors of its constituent words. And, to construct the dense vectors for each news documents, we vectorize each document of the collection with tf-idf based bag-of-words representation. Following this, we select the top-most 20 words as the representative words of each document (i.e., words with the highest tf-idf scores). We then set the dense vector representation of a news document as the sum the constituent word vectors.

To construct the dense vectors for input, we first train the skip-gram model [49] (with the window size parameter of skipgram set to 10) on the 20NG dataset to obtain the dense vectors for each word in the vocabulary. As a next step, we vectorize each document of the collection with tf-idf based bag-of-words representation. Following this, we select the top-most 20 words as the representative words of each document (i.e., words with the highest tf-idf scores). We then set the dense vector representation of a document as the sum the constituent word vectors.

Our approach of obtaining dense representation of the text documents is a relatively simple one instead of more computationally involved approaches such as sequence encoding with LSTMs [165], or contextualized vector representations such as BERT [36]. Since the objective of our experiments is rather to demonstrate the effectiveness of a privacy-aware clustering approach, we keep the input representations relatively simple. However, our methods could be applied on the dense representation of text documents obtained by other recent document encoding methods like BERT, LSTMs etc.

[6] https://github.com/tmikolov/word2vec

Table 4.1: Summary of the datasets used in our clustering experiments.

Dataset	Type	Modality	#Samples	Classify	#Classes
Spiral	Synthetic	2D-Points	312	Points	3
ΛV	Synthetic	2D-Points	373	Points	2
Flame	Synthetic	2D-Points	240	Points	2
MNIST	Real	Image	70K	Digits	10
MNIST-8M	Synthetic	Image	8.1M	Digits	10
Fashion-MNIST	Real	Gray-Image	70K	Fashion-images	10
CIFAR-10	Real	Colour-Image	60K	Images	10
ODPtweets	Real	Text	2.1M	ODP-category	403
20-Newsgroups (20NG)	Real	Text	18K	News-topics	20

4.2 Datasets for Classification Task

A dataset suitable for the purpose of our experiments of “privacy aware supervised learning” needs to be annotated with additional attribute values corresponding to the sensitive information, the prediction of which during the adversarial workflow branch (see Figure 7.2) could then be set up as information leakage. In our supervised learning problem we mainly focused on the classification task (our proposed privacy aware supervised learning is presented in Chapter 7). To test the effectiveness of our proposed approach on different modalities of data, we experiment with both text and image datasets. Details of the datasets used in our supervised classification experiments are provided below and Table 4.2 summarizes these details.

4.2.1 Real Image Datasets

Skin Cancer MNIST (HAM10K):- Contrary to using synthetically generated attribute values for the adversarial task, the ‘Skin Cancer MNIST’ (or HAM10K) dataset [146] allows us to setup the adversarial tasks with two explicitly annotated attributes. The primary task in this dataset involves identifying one out of 7 possible skin diseases, e.g., Bowen’s disease, basal cell carcinoma etc., from images of lesions. The objective in this case is to encode the data in such a way that it does not reveal the age or gender of a person without substantially degrading the effectiveness of the primary task. Some sample images from this ‘Skin Cancer MNIST’ dataset are shown in the Figure 4.5.

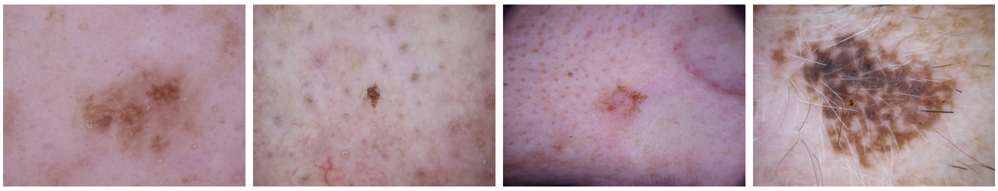


Figure 4.5: Left to right: Lesion images of a young female, mid-aged male, old female and an old male.

4.2.2 Synthetic Image Datasets

Morpho-MNIST (M-MNIST):- The primary task of the original MNIST dataset involves detecting the class of a digit (a gray-scale image with 28×28 pixels) out of the 10 possibilities (one of 0 to 9). As a part of latent information that can potentially be leaked from an encoding of a hand-written image (e.g. a 2d convolution with max-pooling), we first consider the *slant* of a hand-written digit, which can be considered to be correlated with personality traits [28]. To setup the dataset, each *slant label*, z_1 (in our notation), is obtained by applying a threshold on the horizontal shear, α. The value of the shear, α, in turn is computed as a function of second order moments of the gray-scale values, x_{ij} [24]. Formally,

$$z_1 = \begin{cases} 0 & \alpha \leq -0.3 \text{ (left)} \\ 1 & -0.3 < \alpha < 0.3 \text{ (neutral)} \\ 2 & \alpha \geq 0.3.\text{(right)} \end{cases} \tag{4.1}$$

In addition to the slant, the second attribute that we address in our privacy aware supervised learning experiments is whether the image of a hand-written digit is *broken*, i.e., a lack of continuity is exhibited in the strokes. The value of this attribute, if revealed in a real-life situation, could indicate the age of an OCR-ed document to an adversary.

For our experiments with the broken attribute, we use an existing dataset, namely the ‘Morpho-MNIST’, where morphological erosion is applied to synthetically generate broken images [24]. Addition of the synthetically generated broken images, one for each image in the original MNIST, resulted in doubling the number of images for this dataset. The information on whether an image is broken is not available to an adversary, nor does the adversary is allowed to compute the slant labels using Equation

Table 4.2: Summary of the datasets used in our supervised classification experiments.

			#Instances			Primary task		Adversarial Tasks	
Dataset	Type	Modality	Train	Validation	Test	Classify	#Classes	Attribute	Categories
Morpho-MNIST (M-MNIST)	Synthetic	Image	106K	14K	20K	Digits	10	Slant Broken	{left, neutral, right} {yes, no}
Skin Cancer MNIST (HAM10K)	Real	Image	7500	1000	1500	Diseases	7	Age Gender	$\{\leq 30$, 31-60, $> 60\}$ {male, female}
TrustPilot (US English)	Real	Text	20.3K	2700	4K	Sentiment	2	Age Gender	$\{\leq 35, > 35\}$ {male, female}

4.1. Some sample images from this 'Morpho-MNIST' dataset are shown in the Figure 4.6.

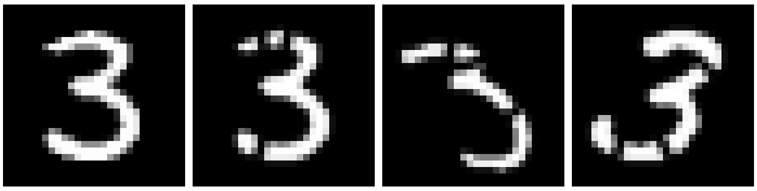


Figure 4.6: Left to right: no slant or broken; no slant but broken; slant on the left and broken; slant on the right and broken.

4.2.3 Real Text Dataset

TrustPilot Dataset:- For the text modality, we use the TrustPilot reviews (the US English subset). The primary task on this dataset involves identifying sentiment (positive or negative) of a review [63]. This dataset, comprised of over $27K$ reviews with sentiment score ranging between 1 and 5, has annotated values for both age and gender. Since the number of reviews with scores 2 and 3 is substantially small, we binarize the sentiment class labels by thresholding with a value of 3, i.e. scores from 1-3 are mapped to class 0 and the rest to 1. Following the previous experiment setup of [34] and [89], we binarize the attribute 'age' as young (age ≤ 35) and its complement (representing the category 'not young'). To construct the dense vector for each review data, which has been used as the input of our 'Privacy-aware supervised classification model', we have applied the same 'skip-gram' model based method described in the Section. 4.1.4 to obtain the dense vector representation of the tweets.

Table 4.3: Summary of the dataset used in our Approximate Nearest Neighbour Search experiments. ρ denotes the population density (#users in a grid cell).

Dataset	#User	#Instances	#Step (τ)	Grid ($\lambda \times \lambda$)	ρ
Traject-10K	10K	1500K	100-200	100×100	149.94
Traject-100K	100K	15M	100-200	1K×1K	14.99
Traject-1M	1M	150M	100-200	1K×1K	149.97
CheckIn-24M	24M	24M	1	107×337	670.85

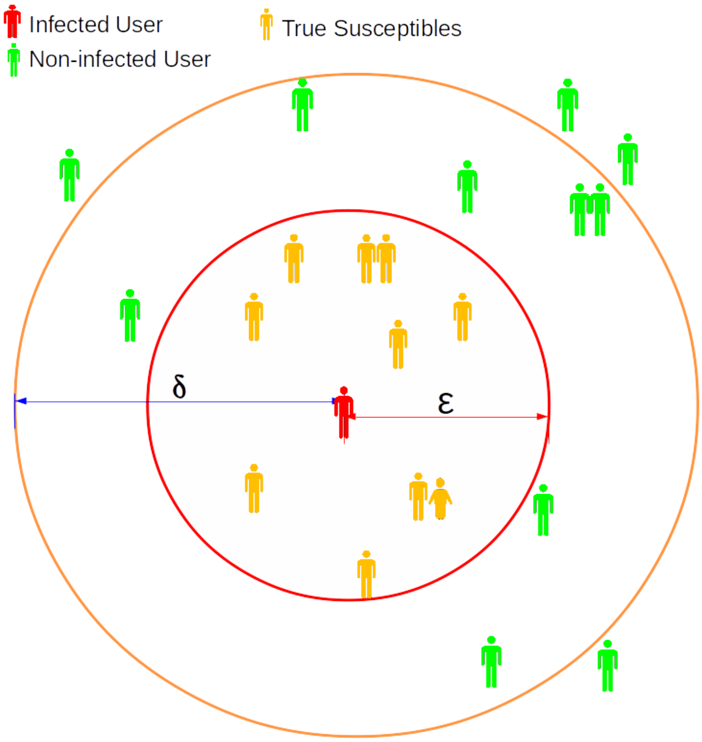


Figure 4.7: Simulated ghost-users (shown in amber color) corresponding to a real infected user (shown in red)

4.3 Datasets for Approximate Nearest Neighbour Search

We performed experimentation of the Approximate Nearest Neighbour Search algorithm in the Chapter 8 on both real and synthetic sample datasets. The FourSquare[7] global check-in dataset has been used as the real dataset in our experiment and for synthetic dataset we have simulated trajectory dataset. Table 4.3 summarizes these datasets [8].

[7] https://drive.google.com/file/d/0BwrgZ-IdrTotZ0U0ZER2ejI3VVk/view

[8] Source code available at https://github.com/chandanbiswas08/infectracer.

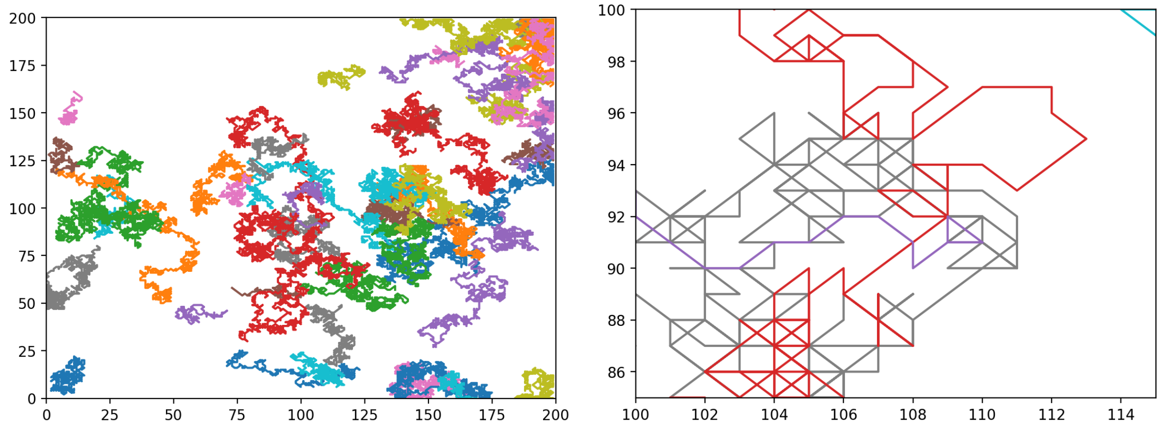


Figure 4.8: *Left*: Random walk based trajectory data of 50 users. *Right*: A zoomed-in view for the trajectory of 3 users. Given the red colored trajectory as a query (infected user) the objective is to retrieve the other two.

Simulated Ground-truths for FourSquare Check-ins

The real FourSquare check-in data is not directly applicable for our study because the data contains only a very small number of simultaneous check-ins of two FourSquare users in the same location (a point-of-interest, e.g. a museum/restaurant). However, to evaluate contact tracing effectiveness under laboratory-settings, our collection requires to contain data for users that came in close contact with each other (in terms of both space and time).

As a solution, we undertake a relatively simple simulation model to generate pseudo-user interactions (likely contacts). First, we filter the original dataset to retain only one check-in per user. This makes the simulation algorithm easier to manage. Next, for each user U (having a unique id), we generate a mutually exclusive set of 'pseudo-users' or 'ghost-users'.

For a user U, as per the generation mechanism, this set of pseudo-users hence represent the ground-truth or the target set of users that need to be retrieved given the current user U as a query. Note that since all the original/real user check-ins were sufficiently apart in space-time coordinates, it is likely that the neighbourhood of a user comprised of the ghost-user check-ins are also far apart (in which case one can rely with sufficient confidence on the simulated ground-truth data).

More concretely, for each user U we generate $p+n$ number of ghost-users in a δ neighbourhood, out of which p belong to an ϵ neighbourhood ($\epsilon < \delta$). If U is infected person then the target is to retrieve the set of p ghost-users. Figure 4.7 presents a visualization of the simulated pseudo-users corresponding to a real infected user (red person in the figure).

As particular values of ϵ and δ, we use 1 and 2. The values of p and n were set to 30 and 60 respectively. The value of n is set to be higher than that of p in order to make the ANN retrieval task more challenging.

Generation of Synthetic Trajectory Dataset

Since the real dataset is limited by the number of available check-ins, in order to collate a larger dataset of locations we generate synthetic data with random walk. Although the real trajectory paths of people are far from being random, the generated data despite being random serves its purpose in the context of our experiments, which is to evaluate the effectiveness of ANN on large volumes of location data.

Since a person is free to move any location, it would acceptable to use synthetic trajectory data, generated by random walk, in our study. Here our task is to identify the suspected person who came close contact to an infected person, so we need to simulate trajectory data which contain location of all user at each time step in some time interval. We use Random walk to generate the synthetic trajectory data in following way.

To generate synthetic data, N simulated agents are initialized each at a randomly chosen location within a 3 dimensional bounding box (each side of the bounding box being in the range λ) with uniform probability. If the location of the i-th agent at timestep t is denoted by (x^i_t, y^i_t, z^i_t), its location x coordinate's value at the next time step is given by

$$x^i_{t+1} = x^i_t + \mathcal{U}(-1,1), \;\; x^i_t \in [0,\lambda] \tag{4.2}$$

and so on for the other spatial dimensions, where $\mathcal{U}(-1,1)$ returns -1 or 1 randomly. The process in Equation 4.2 is repeated for an agent for τ_i number of steps where $\tau_{min} < \tau_i < \tau_{max}$. For our experiments, we used $\tau_{min} = 100$ and $\tau_{max} = 200$. Each generated spatial location for the i-th agent (τ_i number of them in total) is then appended with the time dimension, yielding the set of points of the form

$$L_i = \cup_{t=0}^{\tau_i}\{(x_t^i, y_t^i, z_t^i, t)\}. \tag{4.3}$$

While generating the dataset, at each step, if two agents are found to come sufficiently close to each other, i.e. within an ϵ-neighborhood (ϵ set to 1 similar to the FourSquare dataset settings), we insert each point into the ground-truth (susceptible) list of other. We generate different synthetic datasets with three different values of N (number of simulated agents), namely 10K, 100K and 1M and named with a common prefix 'Traject-' followed by the value of N. Figure 4.8 shows a sample of the generated data with 50 users for the purpose of illustration.

4.4 Evaluation Metrics

The objective of privacy preserving clustering and classification evaluation is to measure the performance of our proposed model under privacy preservation constraint. In this section we discuss about the evaluation metrics used for the purpose of laboratory evaluation of our proposed clustering and classification methods.

4.4.1 Clustering Evaluation Metrics

To measure the effectiveness of our proposed clustering method, we use standard clustering evaluation metrics. Each dataset, that we experiment with, comprises the ground-truth information of class (cluster) labels. As an evaluation metric, we report Normalized Mutual Information (NMI) [35], which measures how homogeneous the clusters are. A different type of clustering effectiveness measure is aggregation of classification results over pairs of data points, yielding higher values if a pair of data points from the same class (in the ground-truth) are predicted to belong to the same cluster. We thus also report F-score and adjusted rand index (ARI) [151] aggregated over these pairwise grouping decisions. Now, we briefly discuss about these metrics.

ARI. The adjusted rand index (ARI) measure appears to be one of the most popular alternatives for comparing partitions. The mathematical formulation of the ARI is as follows.

Let C and C' be the ground truth label and clustering label. Then, the unadjusted rand index (RI) is given by,

$$RI = \frac{a+b}{\binom{N}{2}} \tag{4.4}$$

where, 'a' be the number of pair of elements that are in the same cluster in C and in the same cluster in C' and 'b' be the number of pairs of elements that are in different cluster in C and in different cluster in C'.

Now, the adjusted rand index (ARI) is defined as,

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{4.5}$$

F-score. Let P and Q be respectively the ground truth partition and the predicted partition obtained from a clustering algorithm on the dataset D. Define $Pairs_D$ as the set of all possible pairs of elements of D. Similarly, $Pairs_P$, $Pairs_Q$ are defined as the set of clustered member pairs i.e.,

$$\begin{aligned} Pairs_P &= \{(p_i, p_j) : p_i, p_j \in P_k\} \\ Pairs_Q &= \{(q_i, q_j) : q_i, q_j \in Q_k\} \end{aligned} \tag{4.6}$$

where, P_k and Q_k are an arbitrary cluster in P and Q respectively.

We have defined a contingency matrix for precision-recall in the Table 4.4. Various symbols used in this table are defined as follows.

$$\begin{aligned} tp &= |Pairs_P \cap Pairs_Q|, \\ fp &= |Pairs_Q \setminus Pairs_P|, \\ fn &= |Pairs_P \setminus Pairs_Q| \; and \\ tn &= |Pairs_D \setminus (Pairs_P \cap Pairs_Q)| \end{aligned} \tag{4.7}$$

Table 4.4: Contingency table for precision-recall.

	Pairs in P	Pairs not in P
Pairs in Q	True positive (tp)	False positive (fp)
Pairs not in Q	False negative (fn)	True negatives (tn)

Now, the F-score is defined as,

$$\text{F-score} = \frac{1}{\alpha \frac{1}{Precision} + (1-\alpha)\frac{1}{Recall}} \tag{4.8}$$

where, $Precision = \frac{tp}{(tp+fp)}$ and $Recall = \frac{tp}{(fp+fn)}$.

NMI. Another popular mutual information based measure for evaluating the clustering performance is normalized mutual information (NMI).

Let C and C' be the ground truth label and the label obtained from a clustering algorithm respectively. Then the mutual information between C and C' is given by,

$$MI(C, C') = \sum_{c_i \in C} \sum_{c'_j \in C'} P(c_i, c'_j) log \frac{P(c_i, c'_j)}{P(c_i)P(c'_j)} \tag{4.9}$$

where, $P(c_i)$, $P(c'_j)$ are the probability of an arbitrarily selected sample belongs to the cluster c_i, c_j respectively and $P(c_i, c'_j)$ be the probability of a randomly selected sample belongs to the both clusters c_i and c_j.

The normalized mutual information (NMI) is defined as,

$$NMI(C, C') = \frac{MI(C, C')}{mean(H(C), H(C'))} \tag{4.10}$$

where, $H(C) = \sum_{c_i \in C} P(c_i) log(P(c_i))$ and $H(C') = \sum_{c'_j \in C'} P(c'_j) log(P(c'_j))$ are the entropies of C and C' respectively.

4.4.2 Classification Evaluation Metrics

Accuracy. The effectiveness of a classification model can be measured by 'accuracy'. Informally speaking, the accuracy of a classification model is the fraction of predictions for which the model classify correctly. More formally,

$$Accuracy = \frac{\text{Number of correction predictions}}{\text{Total number of predictions}} \tag{4.11}$$

McNemar's test. McNemar's test is a statistical test [103] used to evaluate the significance of differences in performance between two classifiers or models that are applied to the same dataset. It is specifically designed for paired data, where each instance is classified by both models, and the results are compared to see if there is a significant difference in their performance.

The steps for performing the McNemar's test are following:

Step 1: Create a 2×2 contingency table that represents the classification results of the two models as shown in the Table 4.5.

	Model 2 correct	Model 2 incorrect
Model 1 correct	a	b
Model 1 incorrect	c	d

Table 4.5: Contingency table for the outcome of two models

where,

a = number of instances that both models correctly classified.

b = number of instances that Model 1 classified correctly but Model 2 did not.

c = number of instances that Model 1 misclassified but Model 2 did not.

d = number of instances that both models misclassified.

Step 2: Calculate the McNemar's test statistic which is given by:

$$\chi^2 = \frac{(b-c)^2}{(b+c)} \tag{4.12}$$

Step 3: Compare the test statistic to the χ^2 distribution with a null hypothesis that there is no significant difference between the two models, we can compare the calculated χ^2 value to the χ^2 distribution with 1 degree of freedom (df=1) at a chosen significance level (e.g., 0.05).

If the calculated χ^2 value is greater than the critical value from the χ^2 distribution at the specified significance level, we reject the null hypothesis, suggesting that there is a significant difference in performance between the two models on the given dataset. If the null hypothesis is not rejected, it indicates that there is no significant difference, and the classifiers have similar performance.

4.5 Conclusions

This chapter has presented brief descriptions of various datasets used in the experimentations performed for the present study. It includes synthetically generated datasets as well as real-life sample datasets. Among the latter ones, there are both image datasets and text datasets. We have first described real image datasets such as MNIST, Fashion-MNIST, CIFAR-10, etc. and real text datasets such as ODPtweets Twitter dataset, 20-Newsgroups datasets, etc. which have been used in our experiments of the clustering techniques. Next, we have provided some details of various datasets used in the experimentations of classification methods and approximate nearest neighbour search technique which include Skin Cancer MNIST (HAM10K), Morpho-MNIST (M-MNIST), Trustpilot Reviews, etc. Finally, it has presented various metrics used in our experimentations for evaluation purposes. In the next few successive chapters, we shall explore the research questions introduced in Chapter 1 of this thesis.

Chapter 5

Privacy Aware Unsupervised Learning

[*]

This chapter addresses the first research question RQ-1 introduced in Chapter 1, which is

"*How unsupervised learning algorithm can be re-designed under the constraint of privacy preservation to improve the learning effectiveness?*".

Unsupervised learning plays an important role in the recent advancement of machine learning. Among the various unsupervised learning algorithm K-means is the most popular algorithm due to its simplicity and effectiveness. So, in our first research in this thesis we have worked on the K-means clustering algorithm as an instance of unsupervised learning algorithm. In Chapter 3, we have discussed about the standard K-means algorithm, where we saw that the standard K-means is designed to work well in the real Euclidean space. But to address the research question RQ-1 we need to apply the K-means on Hamming space, which requires some additional consideration of K-means to working properly on Hamming space and hence, we have proposed a novel centroid re-computation procedure of the K-means algorithm. Thus, in this chapter we proposed a privacy preserving approximate K-means clustering algorithm and present a number of experimental results conducted on image and text data to show the effectiveness of our proposed algorithm.

The rest of the chapter is organized as follows. We start with a brief introduction

[*] Some material from [19] has been reused in this chapter.

of privacy preserving clustering in Section 5.1. After that in Section 5.2 we formally introduces the concepts that are used to estimate the centroids during K-means iterations under the privacy preservation constraint. Section 5.3 describes our proposed method that uses mixture of Gaussian based centroid estimation from a set of encoded vectors and global statistics on the input data. In Section 5.4 we present the experiment setup, namely baselines, parameter tuning, and evaluation metrics. This is followed by the presentation of the results of our experiments on synthetic and real datasets (images and text) in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.1 Introduction

Modern advances in software engineering have led to deploying software as services (known as SaaS), which provides an important advantage to organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Consider for example, a 'big-data' clustering SaaS, which takes as input a set of data instances, performs the computations for data clustering on the server side, and returns as output a partitioning of the data to the client.

However, this ubiquitous use of service oriented computational architecture may lead to leakage of information from the input data that a client needs to send to a SaaS component. This information leakage may happen either due to eavesdropping activities in the network or due to malware executed on the servers with intentions of stealing information from the input data. Even when the data appears to be seemingly anonymous with suppressed sensitive information, intelligent processing of the data can reveal sensitive information, such as the infamous *AOL search query data scandal* [105] which exposed the personal identity, or the case of revealing the identities of authors with the help of stylometric features [158].

A solution to preserve data integrity is to *encode* the data in a way that it becomes difficult for any information stealing malware to detect the sensitive information from it. For example, existing literature in differential privacy has proposed a range of approaches for data protection, ranging from pseudo-anonymization of data [102], to adding noise to the data for protecting author information [158] (see [104] for a survey). Each such data-protection initiated transformation needs to achieve a trade-off between two objectives:

i) Ensure that attacks on the encoded data have low likelihood of success, and

ii) The quality of the final output does not change remarkably as a result of the transformation.

In this work, we focus our attention on the latter objective, i.e., ensuring that the output obtained on processing the non-encoded data is not considerably different than the one obtained after encoding the input. The problem, that we particularly focus on, is that of clustering a given set of input vectors. In contrast to assuming a structured form of the input in terms of a database of attribute value lists, as common in existing research on differential privacy focusing on the effectiveness of data protection approaches against deanonymization attacks (see e.g. [65, 150, 102]), we rather focus on a general form of input (real-valued vectors), similar to [158].

In our present approach, we employ a Hamming space transformation of the real-valued data, i.e. we apply a function $\phi : \mathbf{w} \in \mathbb{R}^p \mapsto \mathbf{h} \in \mathbb{H}^m$ to transform every p dimensional real-valued input data vector, $\mathbf{v}$, to a binary vector, $\mathbf{h}$, of m bits. We apply the Super-Bit LSH algorithm as the Hamming space transformation function ϕ, which is given by,

$$\mathbf{h}_i = \text{sgn}(\mathbf{w} \cdot \mathbf{b}_i), \;\; i = 1, \ldots, m, \tag{5.1}$$

where $\mathbf{b}_i$ represents the i^{th} basis vector among m randomly chosen basis vectors followed by normalization and $\text{sgn}(\cdot)$ is the sign function that returns 0 for a negative value of the parameter, and 1 otherwise. Detail description about the Hamming space transformation is presented in the Section 3.2.

The main advantage of the binary transformation, in particular, is that it enables much faster transmission of the data over the network and processing of the data on the server side. This is because it requires only $m/8$ bytes to store a binary vector of m bits, whereas storing a p dimensional real vector requires at least $p \times 4$ bytes of memory.

Next, after encoding the data, we focus on the problem of K-means clustering on this encoded data. Since it is known that a general class of binary transformation functions of real-valued data is a lossy transformation [161, 6], it is important to modify the K-means clustering algorithm with an objective to make it work well with incomplete information. Indeed, this forms the core of our research in this chapter, where we propose a modified K-means algorithm which works under an imposed privacy preservation constraint that it can access only the encoded input. This is in contrast to existing research on fast approximate K-means approaches (see e.g. [131, 75]) which make use of the encoded data vectors in addition to the original ones during different stages of K-means execution.

Our main contribution of this research work in the present chapter is a modified K-means algorithm that respects the privacy preservation constraint, which we call *PPK-means* (privacy preserving K-means) [2] . The constraint makes it imperative to devise an effective method to estimate the centroid vectors during K-means iterations with the incomplete information from the binary encoded input data vectors. Informally speaking, the closer the estimated centroid vectors will be to the true centroids (computed with the complete information from the non-encoded data vectors without the privacy preservation constraint), potentially better will be the output of the clustering algorithm. To this end, we propose a Gaussian mixture model based solution to estimate the bit values of the centroid vectors during the intermediate computational steps. For more reliable estimation of the centroid vectors, we make available for the purpose of computation additional information in the form of aggregated statistics of projected values of the data vectors along a set of randomly chosen basis vectors.

We evaluate our proposed method on a set of both synthetic and real datasets. In comparison to standard K-means, our proposed method, PPK-means, shows significant improvements in terms of latency, without markedly decreasing the clustering effectiveness. Further, our proposed method outperforms the standard K-Means algorithm for clustering a large collection of short documents (tweets).

[2] A prototype of the implementation of PPK-means is available for research purposes at https://github.com/gdebasis/superbit-kmeans.

5.2 Computation of Cluster Centroids

5.2.1 Vector Sum for Centroid Computation

The K centroid vectors during an iteration of K-means algorithm in the Euclidean space of data vectors is given by

$$\mathbf{c^k} = \frac{1}{|W^k|} \sum_{\mathbf{w} \in W^k} \mathbf{w}, \ \ k = 1, \ldots, K \tag{5.2}$$

where W^k denotes the set of vectors in the k^{th} partition. Note that the true computation of the centroid vectors involves making use of the true data points w's.

Under privacy preservation constraints, the true data vectors w's are not available. A way to compute the centroid vectors under privacy preservation constraints is thus to compute centroids in the Hamming space. Formally speaking, the Hamming space represents a modulo 2 finite field (commonly denoted as $GF(2)$), where the (closure ensuring) sum operation is defined as

$$\mathbf{x} \oplus \mathbf{y} = \mathbf{z}, \ \ \text{where} \ \ z_i = (x_i + y_i) \bmod 2 \ \in \{0, 1\}. \tag{5.3}$$

With this definition of the sum operator, the centroid vector in the Hamming space can be computed as

$$\mathbf{h}^k = \bigoplus_{\mathbf{x} \in X^k} \mathbf{x}, \tag{5.4}$$

where the i^{th} component of the vector $\mathbf{h}^k$, denoted by $\mathbf{h}^k_i \in \{0, 1\}$, is given as

$$\mathbf{h}^k_i = (\sum_{\mathbf{x} \in X^k} \mathbf{x}_i) \bmod 2, \tag{5.5}$$

where X^k denotes the set of vectors in the k^{th} partition of the Hamming space of encoded data vectors.

This way of computing the centroids of real-valued vectors, transformed (encoded) in the Hamming space is not optimal because of the apparent inconsistencies in the properties of the transformation function (Equation 5.1) and the modulo 2 addition. To illustrate with an example, consider adding the i^{th} components of two binary vectors both of which are 1, i.e. in other words, the corresponding true data vectors in the Euclidean space yield positive projection values over the i^{th} basis. The projection of the sum vector (in the true data space $\mathbb{R}^p$) over the i^{th} basis must then also be positive, and indeed the i^{th} component of the binary vector (in the JL transformed Hamming space) for the sum must also be encoded as '1' (as per Equation 3.3). More formally, due to the distributional property of the vector addition operation in Euclidean space,

$$\begin{aligned}(\mathbf{w}+\mathbf{v}).\mathbf{b}_i &= \mathbf{w}.\mathbf{b}_i + \mathbf{v}.\mathbf{b}_i \\ &> 0 \text{ if } \mathbf{w}.\mathbf{b}_i > 0 \ \wedge \ \mathbf{v}.\mathbf{b}_i > 0.\end{aligned} \tag{5.6}$$

However, since the value of $(1+1)$ mod 2 is 0, the vector sum of the encoded representations of w and v in the Hamming space produces an output of 0 in the i^{th} component.

5.2.2 Estimation of Optimal Centroids

Given that modulo 2 addition in the Hamming space is problematic, there needs to be an alternate aggregation function to compute the centroid vector in the Hamming space. Moreover, due to the privacy preservation settings, it is not feasible to compute the centroid in the Euclidean space and then transform it to a point in the Hamming space. Therefore, under privacy preservation settings, the only way to compute the Hamming space centroid vectors would be to estimate these values probabilstically with incomplete information rather than computing them deterministically.

Considering the transformation function ϕ, this aggregate function equates to a sum of the signs of the projected values.

$$\begin{aligned}\mathbf{h}_i^k &= 1 \text{ if } \sum_{\mathbf{w}\in W^k} \operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) \geq 0 \\ &= 0 \text{ otherwise}\end{aligned} \tag{5.7}$$

where $\operatorname{sgn}(\mathbf{w}.\mathbf{b}_i)$ returns 1 if $\mathbf{w}.\mathbf{b}_i \geq 0$ and 0 otherwise.

Although the vectors w's in Equation 5.7 are not known due to privacy constraints, the projected values themselves or the signs of these values may be considered to be made available to the server for the purpose of computation without posing a major security threat. Privacy in this case is preserved from the well-known property of locality preserving property of JL lemma that devising an inverse function of ϕ is computationally intractable [6, 161].

The intuition behind estimating the value at i^{th} signature bit of sum vector is that the sum of a large number of positive projected values with a relatively smaller number of negative values is likely to yield a positive result due to the outweighing effect.

In addition to the frequency of the positive projections, their average magnitude values and the skewness of these values can also affect the likelihood of the sum being positive. To model these factors formally, we make use of the Gaussian mixture model (GMM) to estimate the likelihood of the i^{th} bit of the sum vector to be 1.

5.3 Centroid Estimation by Gaussian

5.3.1 Global Distribution of the Projections

Let the set of projected values along the i^{th} basis vector be

$$\mathscr{B}_i = \bigcup_{\mathbf{w} \in W} \mathbf{w}.\mathbf{b}_i. \tag{5.8}$$

We split the set $\mathscr{B}_i$ in two parts according to whether the projection values are positive or negative and assume that the values in each set are generated by a normal distribution, i.e.,

$$\begin{aligned}
&\mathscr{B}_i = \mathscr{P}_i \cup \mathscr{N}_i, \text{ such that} \\
&\mathscr{P}_i = \{\mathbf{w}.\mathbf{b}_i | \mathbf{w}.\mathbf{b}_i \geq 0\}, \ \mathbf{w}.\mathbf{b}_i \sim \mathcal{N}(\mu_i^+, \sigma_i^+) \\
&\mathscr{N}_i = \{\mathbf{w}.\mathbf{b}_i | \mathbf{w}.\mathbf{b}_i < 0\}, \ \mathbf{w}.\mathbf{b}_i \sim \mathcal{N}(\mu_i^-, \sigma_i^-),
\end{aligned} \tag{5.9}$$

where μ_i^+ (μ_i^-) and σ_i^+ (σ_i^-) denote the mean and variance of the positive (negative) projections along the i^{th} basis vector respectively and $\mathcal{N}(\mu, \sigma)$ denotes the Normal distribution with mean μ and variance σ. The parameters of the normal distributions corresponding to each basis vector are computed from the observed projection values, e.g. μ_i^+ and σ_i^+ are computed from the $\mathscr{P}_i$ values.

5.3.2 Distribution of the Sum

During each iteration, a privacy preserving K-means algorithm needs to assign the i^{th} component (bit) of the Hamming vector corresponding to the centroid (vector sum) of the k^{th} partition, h_i^k, to the value of 1 or 0. This binary classification problem thus involves estimating the value of the sum of a set of projection variables (some positive and some negative). We assume that the positive and negative projections (encoded as 1's and 0's respectively) are drawn from two separate distributions. We are interested in the underlying distribution of the sum of these variables. In order to estimate the sum, we present the well known theorem (Theorem 1) that the sum of two normally distributed random variables is also normal.

Theorem 1. *If $Y_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ then the sum of these random variables $Y = Y_1 + Y_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.*

It is easy to prove Theorem 1 using the characteristic function of Normal distributions; for a proof the reader is referred to [44].

In the context of our problem, we assume that the sum of the projected values along i^{th} basis vector corresponding to an arbitrary partition is drawn from the sums of the $\mathscr{P}_i$ and the $\mathscr{N}_i$ values. This value, say x, according to Theorem 1, then follows the distribution

$$x \sim \mathcal{N}\Big(\mu_i^+ + \mu_i^-, (\sigma_i^+)^2 + (\sigma_i^-)^2\Big). \tag{5.10}$$

A value sampled from the distribution of Equation 5.10 is our best guess for the sum of an arbitrary number of reals representing the i^{th} component of the centroid in $\mathbb{H}^m$ belonging to a partition.

5.3.3 Centroid Estimation with Priors from Partitions and Single-component Gaussian

Next, we need to classify the sampled value x into one of the classes (i.e. 1 or 0) for a current partition of the encoded vectors. We leverage the following two sources of information from the observed encoded vectors in each partition to estimate the likelihood of the i^{th} bit of the sum vector in each partition to be 1 (the likelihood of the bit to be set to 0 represents the complementary event).

1. **Hypothesis 1:** If the number of positive projections in a partition contributing to the sum (i.e. the number of vectors with the i^{th} bit observed to be 1) is considerably higher than the number of negative projections, then there is a considerable likelihood of the corresponding bit of the sum vector to be 1.

2. **Hypothesis 2:** If the average of positive projections (over entire dataset) along the i^{th} basis vector is considerably higher than the average over the negative ones, then there is a strong likelihood of the i^{th} bit of the sum of vectors in any partition to be 1.

Using the terminology that B_i^k refers to the set of observed signs of projected values (encoded bit representations), i.e.,

$$\begin{aligned} B_i^k &= \bigcup_{\mathbf{w} \in W_k} \operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) = P_i^k \cup N_i^k \\ P_i^k &= \{\operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) | \operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) \geq 0\}, \mathbf{w} \in W_k \\ N_i^k &= \{\operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) | \operatorname{sgn}(\mathbf{w}.\mathbf{b}_i) < 0\}, \mathbf{w} \in W_k, \end{aligned} \tag{5.11}$$

we estimate the prior probability of the positive class (probability of the i^{th} bit being set to 1) in the k^{th} partition as

$$Pr(\mathbf{h}_i^k = 1 | B_i^k) = \frac{|P_i^k|}{|B_i^k|}. \tag{5.12}$$

A problem with this maximum likelihood priors is that it does not take into account the relative magnitudes of the average values of the positive and negative projections. To this end, we need to address two events in the sampling process - the first of selecting a component (either positive or negative) by observing the respective counts in the partition, and the second, of sampling a value from that component. Stating this formally, the probability of the i^{th} centroid bit being set to 1 (the positive class) is given by

$$Pr(\mathbf{h}_i^k = 1 | B_i^k) = \frac{|P_i^k|}{|B_i^k|} \mathcal{N}\Big(x | \mu_i^+, \sigma_i^+\Big), \tag{5.13}$$

where the variable x represents a sample drawn from Equation 5.10.

5.3.4 Centroid Estimation with Multi-component Gaussian

In Section 5.3.3 we described centroid estimation using Gaussian mixture model (GMM) with two components corresponding to the positive and negative projection values. In this section, we generalize the idea further by defining multiple components for positive and negative projections.

Motivation

GMM with multiple components may model substantial differences between the projection values of the same sign. With a binary GMM, the only parameter that can handle these differences is the variance parameter σ_i^+ (or σ_i^- for the negative projections). However, a multiple number of components, where each component generates projected values of the same sign (either positive or negative) within specific ranges, gives an estimate about the magnitude of the values, as opposed to estimating only their differences from the average (for the binary case). This estimate about the magnitude may potentially result in improving the estimate for the sign of $\mathbf{h}_i^k$, where the absolute value of a sum of a small number of projections could be higher than those of a much larger number of projections of the opposite sign.

Formal Description

To enable a more fine-grained approach to count the priors and the posteriors, we assume that the set of projected values follow a multi-component Gaussian mixture model, where values within a specific range are assumed to be generated from one particular component of the Gaussian mixture. In our approach, we divide the positive and the negative projected values into a number of (M a parameter) equal length intervals. More specifically, we store the global statistics of the projected values along each dimension i as

$$\begin{aligned}
&\mathscr{B}_i = (\cup_{j=1}^{M} \mathscr{P}_i^j) \cup (\cup_{j=1}^{M} \mathscr{N}_i^j), \ \text{such that} \\
&\mathscr{P}_i^j = \{\mathbf{w}.\mathbf{b}_i | j\delta_i^+ \leq \mathbf{w}.\mathbf{b}_i < (j+1)\delta_i^+\}, \ \mathbf{w}.\mathbf{b}_i \sim \mathcal{N}(\mu_i^{j+}, \sigma_i^{j+}) \qquad (5.14) \\
&\mathscr{N}_i^j = \{\mathbf{w}.\mathbf{b}_i | j\delta_i^- \leq \mathbf{w}.\mathbf{b}_i < (j+1)\delta_i^-\}, \ \mathbf{w}.\mathbf{b}_i \sim \mathcal{N}(\mu_i^{j-}, \sigma_i^{j-}),
\end{aligned}$$

where each $\mathscr{P}_i^j$ ($\mathscr{N}_i^j$) represents a Gaussian generating positive (negative) projection values of the points within the j^{th} interval ($j = 1, \ldots, M$), μ_i^{j+} (μ_i^{j-}) and σ_i^{j+} (σ_i^{j-}), respectively, refer to the mean and the variance of the positive (negative) projected values within the j^{th} interval, and δ_i^+ (δ_i^-) represents the length of each positive (negative) intervals in i^{th} dimension, computed as

$$\delta_i^+ = \frac{(\mathbf{w}.\mathbf{b}_i)_{max} - (\mathbf{w}.\mathbf{b}_i)_{min}}{M}, \ \forall \mathbf{w}.\mathbf{b}_i \geq 0. \tag{5.15}$$

Similar to the binary case of Equation 5.10, to obtain the distribution of the sum, we sample a likely value of the projection of the sum vector from the distribution

$$x \sim \mathcal{N}\Big(\sum_{j=1}^{M}(\mu_i^{j+} + \mu_i^{j-}), \sum_{j=1}^{M}((\sigma_i^{j+})^2 + (\sigma_i^{j-})^2)\Big). \tag{5.16}$$

During clustering, let z denote the latent variable indicating the component from which the sum of the projection along the i^{th} dimension (denoted by x in Equation 5.16) is most likely to be sampled from. Using uniform priors, the maximum likelihood value of this latent variable is then estimated as ζ^+ when $x \geq 0$ and ζ^- otherwise. Mathematically,

$$\begin{aligned} \zeta^+ &= \arg\max_{j=1}^{M} \mathcal{N}\Big(x|\mu_i^{j^+}, \sigma_i^{j^+}\Big), \text{ if } x \geq 0, \\ \zeta^- &= \arg\max_{j=1}^{M} \mathcal{N}\Big(x|\mu_i^{j^-}, \sigma_i^{j^-}\Big), \text{ if } x < 0, \end{aligned} \tag{5.17}$$

That is, we use $\mathcal{N}(\mu_i^{j^+}, \sigma_i^{j^+})$'s as the posteriors when $x \geq 0$ and $\mathcal{N}(\mu_i^{j^-}, \sigma_i^{j^-})$'s otherwise. Next, after estimating the values of $z = \zeta^+$ (or ζ^-), we compute the likelihood of $\mathbf{h}_i^k$ by using the local priors (similar to Equation 5.13) with the help of Equation 5.18.

$$\begin{aligned} Pr(\mathbf{h}_i^k = 1|B_i^k, z) &= \frac{|P_i^k|}{|B^k|}\mathcal{N}\Big(x|\mu_i^{\zeta^+}, \sigma_i^{\zeta^+}\Big), \text{ if } x \geq 0 \\ Pr(\mathbf{h}_i^k = 0|B_i^k, z) &= \frac{|N_i^k|}{|B^k|}\mathcal{N}\Big(x|\mu_i^{\zeta^-}, \sigma_i^{\zeta^-}\Big), \text{ if } x < 0 \end{aligned} \tag{5.18}$$

where, P_i^k and B_i^k are defined as per Equation 5.11.

The multi-component case of Equation 5.18 is a generalization of the binary component case (Equation 5.13), the generalization ensuring that the posteriors are estimated over a small (and hence more reliable) range of values. It is to be noted that multiple components only apply to the posteriors and not to the local priors of each cluster which are still binary as per the definition of Equation 5.11.

Algorithm 1: Client: Hamming space Transformation

```
Input: $X = \{\mathbf{x}\}$: A collection of vectors, $\mathbf{x} \in \mathbb{R}^p$
Input: $m$: Hamming code length
Input: $M$: # GMM components, 0 for PPK-means with priors-only
Output: $\mu_i^{j+}$ ($\mu_i^{j-}$): Means of positive (negative) projections w.r.t. the $i^{\text{th}}$ basis vector ($i = 1, \ldots, m$)
        along the $j^{\text{th}}$ positive (negative) GMM component ($j = 1, \ldots, M$)
Output: $\sigma_i^{j+}$ ($\sigma_i^{j-}$): Variances of positive (negative) projections w.r.t. the $i^{\text{th}}$ basis vector ($i = 1, \ldots, m$)
        along the $j^{\text{th}}$ positive (negative) component ($j = 1, \ldots, M$)
Output: $X' = \{\mathbf{h} : \mathbf{h} \in \mathbb{H}^m\}$: A transformed set of Hamming vectors
begin
    Select basis vectors $\mathfrak{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ as in Super-Bit LSH algorithm [68]
    // Send global statistics to clustering SaaS only if posteriors are
       to be used
    if $M > 0$ then
        $\delta_i^+ \leftarrow \Big(\max_{\mathbf{x} \in X : \mathbf{b}_i^T \cdot \mathbf{x} \geq 0} \mathbf{b}_i^T \cdot \mathbf{x} - \min_{\mathbf{x} \in X : \mathbf{b}_i^T \cdot \mathbf{x} \geq 0} \mathbf{b}_i^T \cdot \mathbf{x}\Big)/M$
        $\delta_i^- \leftarrow \Big(\max_{\mathbf{x} \in X : \mathbf{b}_i^T \cdot \mathbf{x} < 0} \mathbf{b}_i^T \cdot \mathbf{x} - \min_{\mathbf{x} \in X : \mathbf{b}_i^T \cdot \mathbf{x} < 0} \mathbf{b}_i^T \cdot \mathbf{x}\Big)/M$
        $(\mu_i^{j+}, \sigma_i^{j+}) \leftarrow (\mathbb{E}, \mathrm{Var})_{\mathbf{x} \in X : j\delta_i^+ \leq \mathbf{b}_i^T \cdot \mathbf{x} < (j+1)\delta_i^+} \mathbf{b}_i^T \cdot \mathbf{x}$
        $(\mu_i^{j-}, \sigma_i^{j-}) \leftarrow (\mathbb{E}, \mathrm{Var})_{\mathbf{x} \in X : j\delta_i^- \leq \mathbf{b}_i^T \cdot \mathbf{x} < (j+1)\delta_i^-} \mathbf{b}_i^T \cdot \mathbf{x}$
    for each $\mathbf{x} \in X$ do
        for $i = 1, \ldots, m$ do
            $\mathbf{h_i} \leftarrow sgn(\mathbf{b}_i^T \cdot \mathbf{x})$
        $X' \leftarrow X' \cup \mathbf{h}$
```

Detailed working steps of client-side data encoding (including computing the global projection statistics) and server side centroid estimation (mainly involving how to use projection values for better estimation) are presented in Algorithms 1 and 2 .

5.4 Experimental Setup

We conduct experiments to show the effectiveness of our proposed algorithm, namely PPK-means. The objective of our experiments is to investigate:

a) Whether PPK-means with encoded data yields results that are comparable (not remarkably different clustering results) with the standard K-means, which has access to the true data;

b) The best settings of PPK-means, in terms of mainly how many components to use in the GMM and the effects of priors and posteriors; and

c) The run-time efficiency of PPK-means with respect to standard K-means.

Algorithm 2: PPK-means on Clustering SaaS

```
Input: $X$: A transformed set of binary vectors ($\mathbb{H}^m$) received from a client as the output of Algorithm 1
Input: $K$: #desired clusters
Input: $M$: #GMM components, 0 for PPK-means with priors-only
Input: $\mu_i^{j+}$ ($\mu_i^{j-}$) and $\sigma_i^{j+}$ ($\sigma_i^{j-}$): Means and variances of positive (negative) projections w.r.t $j^{\text{th}}$ GMM
        component ($j = 1, \dots, M$) along $i^{\text{th}}$ basis ($i = 1, \dots, m$)
Input: $T$: maximum number of iterations
Output: A $K$-partition of $X$ such that $\bigcup_{k=1}^{K} X^k = X$
begin
    Randomly initialize $K$ cluster centres $\mathbf{h}^1, \dots, \mathbf{h}^K \in X$
    for $t = 1, \dots, T$ do
        // Assign every x to its nearest centroid
        foreach $\mathbf{x} \in X - \bigcup_{k=1}^{K}\{\mathbf{h}^k\}$ do
            $k' \leftarrow \arg\min_k(d_H(\mathbf{x}, \mathbf{h}^k))$ $X^{k'} \leftarrow X^{k'} \cup \mathbf{x}$
        for $k = 1, \dots, K$ do
            // Recompute cluster center
            for $i = 1, \dots, m$ do
                $PosCount \leftarrow 0$
                foreach $\mathbf{x} \in X^k$ do
                    if $\mathbf{x}_i = 1$ then
                        $PosCount \leftarrow PosCount + 1$
                $NegCount \leftarrow |X^k| - PosCount$
                if $M = 0$ then
                    if $rand(0,1) \leq \frac{PosCount}{|X^k|}$ then $\mathbf{h}_i^k = 1$
                    else $\mathbf{h}_i^k = 0$
                else
                    $\alpha \leftarrow \mathcal{N}\Big(\sum_{j=1}^{M}(\mu_i^{j+} + \mu_i^{j-}), \sum_{j=1}^{M}((\sigma_i^{j+})^2 + (\sigma_i^{j-})^2)\Big)$
                    if $\alpha > 0$ then
                        $\zeta^+ \leftarrow \arg\max_{j=1}^{M} \mathcal{N}(x|\mu_i^{j^+}, \sigma_i^{j^+})$
                        $S^+ \leftarrow \frac{PosCount}{|X^k|}\mathcal{N}(x|\mu_i^{\zeta^+}, \sigma_i^{\zeta^+})$
                        if $rand(0,1) \leq S^+$ then $\mathbf{h}_i^k = 1$
                        else $\mathbf{h}_i^k = 0$
                    else
                        $\zeta^- \leftarrow \arg\max_{j=1}^{M} \mathcal{N}(x|\mu_i^{j^-}, \sigma_i^{j^-})$
                        $S^- \leftarrow \frac{NegCount}{|X^k|}\mathcal{N}(x|\mu_i^{\zeta^-}, \sigma_i^{\zeta^-})$
                        if $rand(0,1) \leq S^-$ then $\mathbf{h}_i^k = 0$
                        else $\mathbf{h}_i^k = 1$
```

5.4.1 Datasets

For visual observation we conduct our privacy preserving clustering experiments on the three synthetic 2D points datasets, namely ‘Spiral’, ‘ΛV’ and ‘Flame’. Along with these three datasets we conduct the experiments of our proposed methods and other baseline methods on two large scale image and text datasets, namely ‘MNIST-8M’ (gray-scale image) and ‘ODPtweets’ (text) datasets. The detail description of these synthetic 2D points datasets along with the image and text datasets are presented in the Section 4.1.

5.4.2 Baselines

To test the effectiveness of estimating centroids with incomplete information (privacy preservation settings), we employ a number of baseline K-means clustering methods. Additionally, we also compare our results with the standard K-means, which works with the true data without the privacy preservation constraint. It is to be noted that since the standard K-means is not privacy preserving, instead of treating it as a baseline, it is rather treated as an *apex-line* to get an idea about the best results that could be obtained under ideal settings on a particular dataset.

LSH-based partitioning

Locality sensitive hashing (LSH) is a general class of data compression methods which seek to assign identical hash codes (called signatures) to vectors that are similar to each other. A commonly used LSH algorithm, called the MinHash, involves intersection of random permutations of the components in data [120]. The algorithm proposed in [6] extended MinHash based LSH to real-valued vectors in high dimensions by taking projections with respect to randomly chosen basis vectors. As our first baseline, we use the method proposed in [6] to partition the data into K classes. More specifically, for a given value of K, we compute the LSH signature of each data point ranging from 1 to K and then group together the data points by their binary encoded signature values. This ensures that similar points are clustered together (since they are expected to have similar signatures). In this algorithm, the K-means computation only needs to access the binary encoded signature values, as a result of which it is privacy preserving. We name this baseline approach ‘LSH-partition’.

K-means over Hamming Space

To show the usefulness of centroid estimation, we employ the standard K-means approach that takes as inputs the encoded data, $\mathbf{x} = \phi(\mathbf{w})$, where $\phi : \mathbb{R}^p \mapsto \mathbb{H}^m$ is the Super-Bit LSH encoding function [68] (see Section 3.2). Different to our approach, we perform standard K-means clustering over the continuous space $\mathbb{R}^m$ (instead of considering only the discrete subspace $\mathbb{H}^m$), as a result of which the vector sum operation becomes a closed operation in $\mathbb{R}^m$. Since this baseline does not use a modified vector sum operation for computing the centroids (as PPK-means does with the GMM-based estimation), any errors in the encoding function are likely to propagate and potentially cause considerable differences in results with respect to applying K-means on the original data. Note that we call this baseline 'HK-means' (K-means algorithm on a Hamming space).

K-means convergent on Hamming Space

Similar to HK-means, in this baseline we execute K-means on the encoded set of binary vectors (signatures) in m dimensions. However, instead of treating the embedded space as the extended space of m dimensional real vectors, $\mathbb{R}^m$, we restrict the embedded space to the discrete space $\mathbb{H}^m$. Consequently, the standard notion of the vector sum operation involving component-wise addition is no longer a closed operation in $\mathbb{H}^m$, which requires redefining this operation to be able to execute K-means. Specifically, using the property that $\mathbb{R}^m$ is point-wise convergent, we compute the k^{th} cluster centroid as

$$\mathbf{h}_i^k = \operatorname{sgn}\Big(\frac{1}{|X^k|}\sum_{\mathbf{x}\in X^k} x_i - \frac{1}{2}\Big),\ \mathbf{x} \in \mathbb{H}^m. \tag{5.19}$$

It can be easily verified that the centroid computation of Equation 5.19 is a closed operation, i.e. $\mathbf{h}^k \in \mathbb{H}^m$. Informally speaking, we first compute centroid vectors $\mathbf{h}^k$'s over the extended space $\mathbb{R}^m$, and then to maintain the closure property, we map a centroid $\mathbf{h}^k$ to its nearest point in the Hamming space (a similar approach was used in [107] to modify the skip-gram [49] objective function for obtaining binary embedding of graph nodes).

Since this baseline computes centroids using the Euclidean space and then 'truncates' these centroids to the nearest point in the Hamming space, we call this baseline 'E2HK-means' (Euclidean to Hamming convergent K-means).

5.4.3 Parameters and Evaluation Metrics

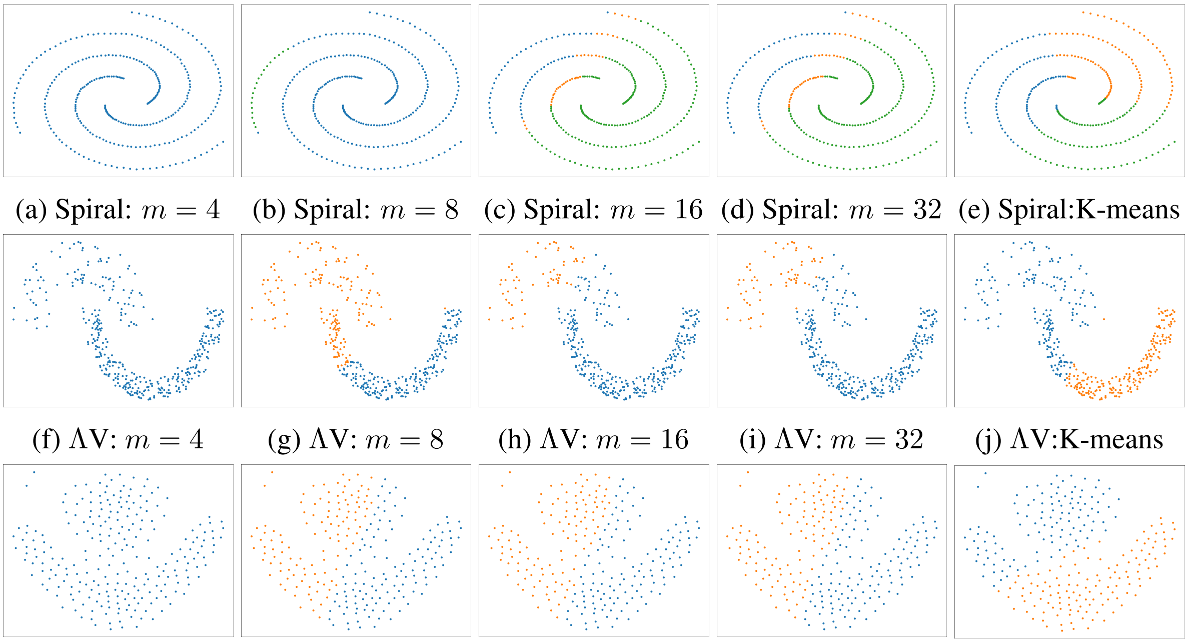
A parameter to PPK-means is the dimensionality of the Hamming space in which the p dimensional data needs to be transformed. Another parameter is the number of components, M, for the GMMs used to estimate the projected values of each sign (positive and negative) in PPK-means. A value of $M = 1$ corresponds to using a Normal distribution each for the positive and negative projections. We also investigate the use of posteriors in combination with the priors (Equation 5.13) vs. the use of priors only (Equation 5.12).

To measure the effectiveness of our proposed method, we use standard clustering metrics. Each dataset, that we experiment with, comprises the ground-truth information of class (cluster) labels. As an evaluation metric, we report Normalized Mutual Information (NMI) (see Section 4.4.1 for description of NMI), which measures how homogeneous the clusters are. A different type of clustering effectiveness measure is aggregation of classification results over pairs of data points, yielding higher values if a pair of data points from the same class (in the ground-truth) are predicted to belong to the same cluster. We thus also report F-score and adjusted rand index (ARI) (see Section 4.4.1 for description of F-score and ARI) aggregated over these pairwise grouping decisions. Additionally, we also measure the efficiency of the clustering methods in terms of computational latency. For a fair comparison of runtimes, all experiments were conducted on a 64-Bit Linux workstation with Intel Xeon 'E5-1620 3.60GHz' CPU and 48 GB RAM.

5.5 Results

5.5.1 Visual Comparison with K-means

To visually investigate the effectiveness of the centroid estimation process of PPK-means, we first report results of PPK-means on the three synthetic 2D points datasets described in Section 4.1.1, and then compare these with the ideal scenario of standard K-means executed on the original (non-encoded) data. It is to be mentioned that we do not report results with the other baselines (outlined in Section 5.4.2) be-



(a) Spiral: $m = 4$ (b) Spiral: $m = 8$ (c) Spiral: $m = 16$ (d) Spiral: $m = 32$ (e) Spiral:K-means

(f) ΛV: $m = 4$ (g) ΛV: $m = 8$ (h) ΛV: $m = 16$ (i) ΛV: $m = 32$ (j) ΛV:K-means

(k) Flame: $m = 4$ (l) Flame: $m = 8$ (m) Flame: $m = 16$ (n) Flame: $m = 32$ (o) Flame:K-means

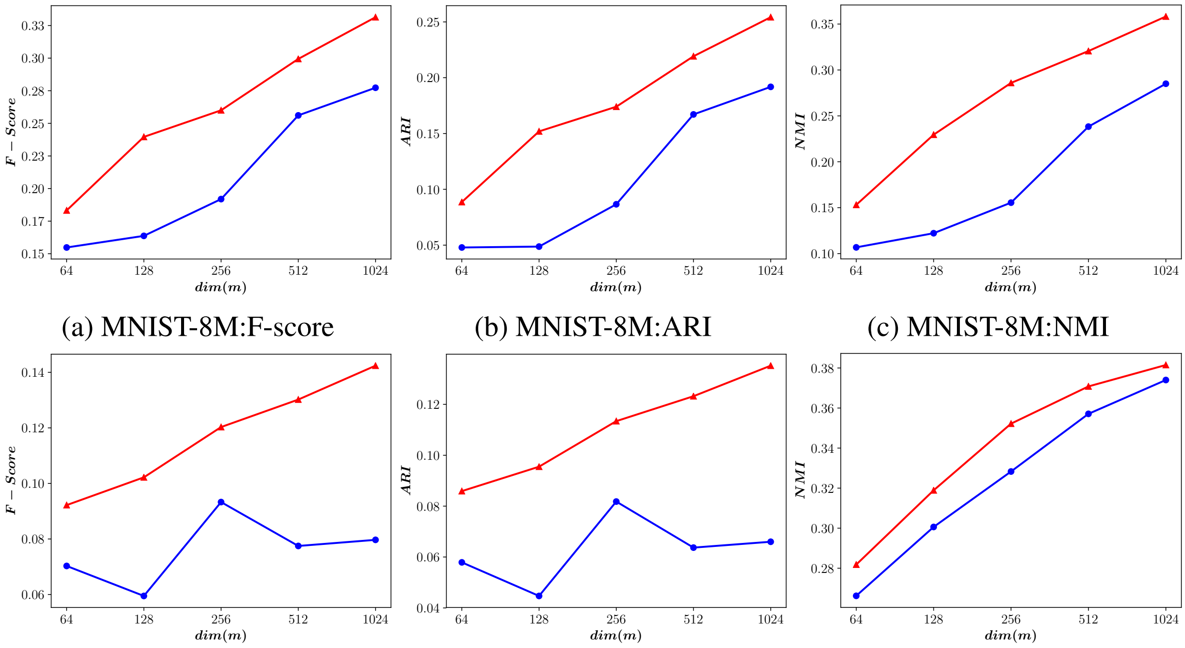
Figure 5.1: Comparison of clusters obtained after 5 iterations of the PPK-means algorithm corresponding to different number of Hamming space encoding dimensions (m): Plots shown in 1^{st}, 2^{nd} and 3^{rd} rows correspond to Spiral, ΛV and Flame datasets respectively. The results of standard K-means algorithm on non-encoded data are shown in the rightmost plot (Figure 5.1e, 5.1j and 5.1o) of the respective rows. Plots shown from left to right barring the rightmost ones correspond to the different values of Hamming space encoding dimension, $m = 2^n$ with $n = 2, 3, 4, 5$ respectively. The value of parameter K for both PPK-means and K-means were set to the number of true clusters (as per the ground-truth). The PPK-means version used for obtaining the plots only involved the priors only (Section 5.3.3).

cause our experiments with the MNIST-8M and ODPtweet dataset already revealed that these baselines resulted in worse clustering effectiveness (see Table 5.1). Further, we also report clustering results for PPK-means with priors-only configuration, because after visual inspection, we noticed that these results were indistinguishable from the ones that used the posteriors.

Figure 5.1 shows the partitions obtained during intermediate steps of executing PPK-means. An interesting observation is that for PPK-means to work well, the dimension of the Hamming space needs to be sufficiently larger than p (the dimension of the original data points). As an extreme case, it can be seen that most points are clustered into a single group with the configuration $m = 4$ on all the datasets. The results improve with $m = 8$ and higher. For the 'Spiral' dataset, $m = 8$ is not able to find out the 3 natural clusters (it finds only 2). It can also be seen that the results with $m = 16$ and $m = 32$ are comparable with those of K-means. This is an important

Table 5.1: Comparison of PPK-means algorithm against baseline clustering approaches on MNIST-8M ($K = 10$) and ODPtweets ($K = 403$) datasets. The value of K (# desired clusters) was set equal to # reference clusters. # iterations for each method was set to 10.

Method	Centroid	$\phi : \mathbb{R}^p \mapsto \mathbb{H}^m$	Privacy	MNIST-8M				ODPtweets			
Name	Estimation	$(m =)$	preserve	F-score	ARI	NMI	Time (s)	F-score	ARI	NMI	Time (s)
LSH-partition [6]	None	1024	True	0.1871	0.0460	0.0817	6664	0.0236	0.0037	0.0936	512
HK-means	$\sum \mathbb{R}^m$ (centroids $\in \mathbb{R}^m$)	1024	True	0.2967	0.2143	0.3012	18782	0.1311	0.1261	0.3790	7492
E2HK-means	$\lim(\sum \mathbb{R}^m) \mapsto \mathbb{H}^m$	1024	True	0.3015	0.2196	0.3307	10669	0.1205	0.1161	**0.3833**	1580
PPK-means	GMM priors only	1024	True	0.2773	0.1918	0.2850	6013	0.0797	0.0659	0.3740	625
PPK-means ($M = 1$)	Single-component GMM	1024	True	0.2812	0.1981	0.2851	13196	0.1200	0.1125	0.3758	1820
PPK-means ($M = 10$)	Multi-component GMM	1024	True	**0.3314**	**0.2542**	**0.3582**	15807	**0.1423**	**0.1351**	0.3815	1860
K-means	$\sum \mathbb{R}^p$	N/A	False	0.3573	0.2852	0.3951	190138	0.1078	0.1015	0.3610	2057



(a) MNIST-8M:F-score (b) MNIST-8M:ARI (c) MNIST-8M:NMI

(d) ODP-Tweet:F-score (e) ODP-Tweet:ARI (f) ODPtweets:NMI

Figure 5.2: Plots of the values (with respect to two datasets MNIST-8M and ODPtweets) of three metrics F-Score, ARI and NMI of PPK-means with priors-only (Blue), PPK-means with multi-component GMM ($M = 10$) (Red) against the Hamming space dimension m ($64 \leq m \leq 1024$) in multiples of CPU word size 16.

observation which shows that PPK-means, even without the complete knowledge of data, can yield comparable results with those of K-means. This shows that the PPK-means can potentially work well as a privacy preserving K-means algorithm.

Table 5.1 shows the results of comparing the performance of two different settings of PPK-means (with and without posteriors) with the three baseline algorithms (as presented in Section 5.4.2) on the MNIST-8M dataset. Firstly, we observe that the LSH-based partition yields poor results in terms of F-score, ARI, and NMI, which shows that it tends to group dissimilar feature instances into the same group, i.e., it classifies most of the digits into a small number of clusters thus resulting in largely non-homogeneous clusters. Multi-component based PPK-means outperforms the other baselines (including the single component K-means), indicating the usefulness of the posteriors and a multiple number of Gaussians to better estimate the centroids of each cluster. In fact, the performance of the proposed multi component PPK-means is seen to be comparable with the standard K-means for the MNIST-8M dataset.

It is worth noting that the execution times of all variants of the proposed method are considerably smaller than that of the standard K-means algorithm. An important implication is that the proposed method (specifically, PPK-means with multi component) achieves comparable performance as standard K-means with remarkably smaller execution time.

Similar trends are also observed for the text dataset. In particular, from Table 5.1, it can be seen that the clustering results of the proposed GMM based methods that make use of the posterior information are impressively better than the baselines (and also standard K-means). While the performance of the proposed methods are comparable to that of HK-means and E2HK-means for some metrics, the optimal performance is observed for the multi-component GMM based PPK-means on both F-score and ARI.

The fact that the effectiveness of PPK-means without using the posteriors and the multiple components is lower (in comparison to the case where we use this information) indicates that

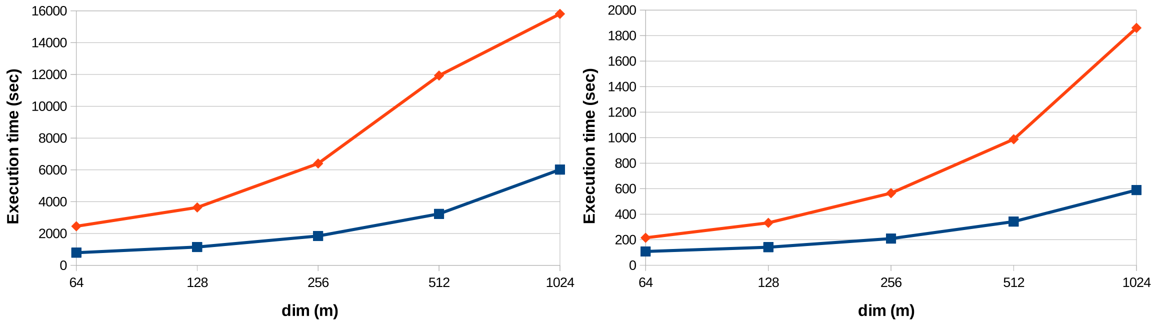
1. Only using the priors in PPK-means may not be able to capture the situation when a small number of positive projected values can dominate the overall sum involving a larger number of negative values or vice-versa.

2. Using a single Normal distribution to model the projected values of a particular sign may not be expressive enough to capture the variations in the projected values themselves.

The above limitations of the priors-only based and the single component GMM based PPK-means are addressed by a) using the posterior information (in the form of global statistics of the averages and the variances of the projected values), and b) by employing a multiple number of intervals to generalize the $M = 1$ case. With $M = 10$ (the best we achieved by varying M within a range of 2 to 10), the PPK-means algorithm is able to better estimate the centroid vectors by using a more fine-grained approach leveraging the additional information about the different ranges of the projected values. Consequently, the estimated centroid vectors are more similar to their true counterparts (i.e. the ones obtained with K-means on non-encoded data and then transformed to the Hamming space).

With respect to run-times, it can be observed that making use of posteriors and multiple components can lead to increase in run-times as opposed to the single component priors-only case. This increase in time can be attributed to the computation and transportation (from a client to the cluster SaaS) of more information, namely the mean and the variances of the projected values. Increasing the number of GMM components (M) also leads to increasing the run-time since the computation then involves estimating the likelihood of the sign of each component of a centroid vector from a multiple number (M) of components.

Another interesting observation about the run-times is that all the baseline approaches and PPK-means execute much faster than the standard K-means without the privacy constraint. This is because the encoded data is stored as integers of 8 bytes (a word size) in the main memory which is much smaller than storing real-valued vectors (e.g. storing 1024 dimensional Hamming vectors requires 1024/64=16 words of memory, whereas storing a 784 dimensional real-valued vector consumes 784 words of memory). Moreover, encoding vectors as integers also leads to much faster inner product based similarity computation between them in comparison to the computationally expensive floating point operations of real-valued vectors, e.g., to compute the similarity between two 1024 dimensional Hamming vectors, one simply needs to execute the `POPCOUNT` machine instruction 16 times (16=1024/64).



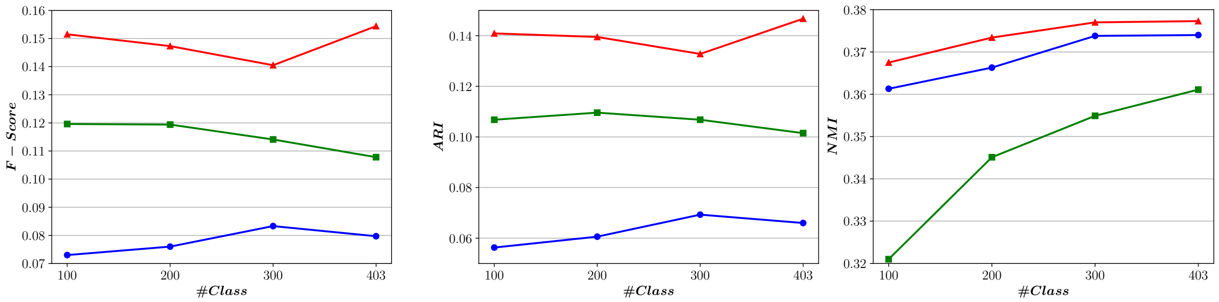
(a) MNIST-8M (b) ODPtweets

Figure 5.3: Plots of computation time of PPK-means with priors-only (Blue) and PPK-means with multi-component GMM ($M = 10$) (Red) versus the Hamming space dimension m ($64 \leq m \leq 1024$) with respect to two different datasets MNIST-8M and ODPtweets.

5.5.2 Parameter Sensitivity Analysis

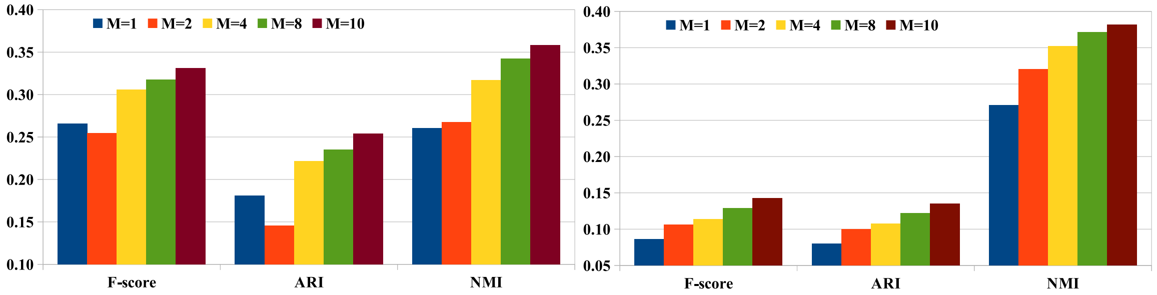
We now investigate the effects of varying the encoding dimension (m) and the number of components for GMM estimation (M) in PPK-means. Figure 5.2 shows the relative differences between PPK-means (priors only) and PPK-means with GMM posteriors ($M = 10$) for different values of m within 64 to 1024, each value of m being a multiple of 16 (CPU word size). From the figure, it can be seen that the downstream clustering effectiveness is proportional to the value of m implying that low dimensional Hamming representations tend to lose information about the original data vectors. It is interesting to see that even with noisy representation of the encoded vectors, GMM posterior-based PPK-means shows substantial differences in results as compared to its prior-only counterpart (see the relatively large differences of F-score, ARI and NMI values). This suggests that the posterior based PPK-means is more robust under parsimonious settings of memory and CPU usage. Figure 5.3 shows that the execution time increases drastically with larger values of m (which was one of the reasons why the value of m was restricted up to 1024 in our experiments).

As the MNIST-8M dataset contains images of the 10 digits (0 to 9), the number of desired clusters (K) is 10. However in many practical scenarios, the ideal number of clusters is not known apriori. To test the robustness of proposed methods without the information about the true number of clusters, we evaluate the clustering effectiveness of the competing methods by varying the value of K. From Figure 5.4, we observe that PPK-means (multi-component GMM) outperforms both standard



(a) F-score (b) ARI (c) NMI

Figure 5.4: Comparative performance of PPK-means with priors-only (Blue), PPK-means with multi-component GMM ($M = 10$) (Red) and K-means (Green) versus the number of desired clusters (K) with respect to three metrics F-Score, ARI and NMI on samples of ODPtweets dataset.

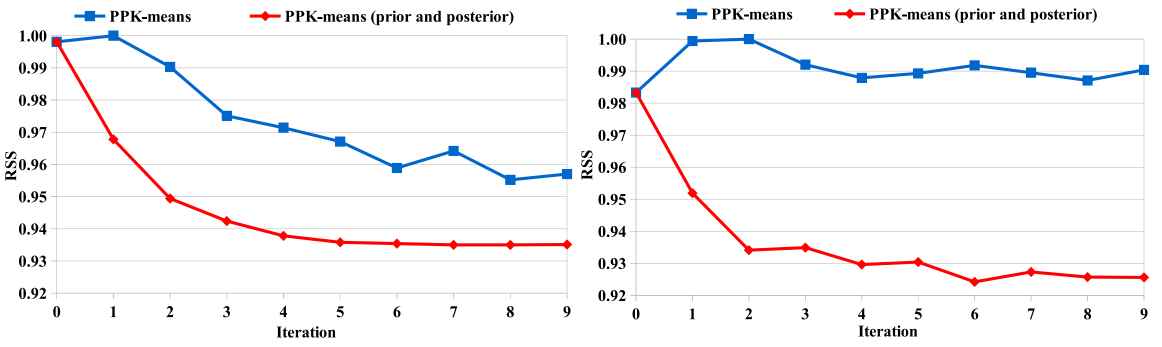


(a) MNIST-8M (b) ODPtweets

Figure 5.5: Comparative performance of PPK-means with multi-component GMM for different values of M with respect to three metrics F-Score, ARI and NMI on the MNIST-8M and ODPtweets datasets.

K-means and the version of PPK-means that uses priors only. In Figure 5.5, we investigate the effect of varying the number of GMM components in PPK-means. It can be seen that increasing M tends to increase clustering effectiveness.

Figure 5.6 plots an intrinsic clustering evaluation metric, namely the residual sum of squares (RSS), which is measured by first aggregating the distances of the constituent points of a cluster from its centroid and then averaging these values over all clusters in the dataset. The smaller the RSS value, the better is the clustering output. Figure 5.6 shows that the multi-component ($M = 10$) GMM setting of PPK-means leads to sharper drops in normalized RSS values across iterations than its priors-only counterpart. The priors only mode of PPK-means can sometimes also lead to increasing the RSS value across iterations (see the increase from iteration 6 to 7), which can happen due to the uncertainties involved in centroid estimation. However, the fact that the RSS values steadily decrease for the posterior mode of PPK-means, shows



(a) MNIST-8M (b) ODPtweets

Figure 5.6: Variations in RSS values for PPK-means with priors-only and PPK-means ($M = 10$).

that the centroid estimations in this case are more robust.

5.6 Conclusions

We investigated the problem of K-means clustering under a privacy preservation constraint. This constraint requires the input data to be sent in an encoded format to a server offering clustering as a ‘software as a service’ (SaaS), such that the data is protected from any information leaking threats (e.g. deanonymization and authorship attribution). We propose a modified K-means algorithm, called PPK-means, that leverages additional pieces of information, e.g. global statistics on the projected values of the original data vectors along random basis vectors used for the purpose of encoding. Experimentation on image and textual data demonstrates that the proposed approach, by leveraging information in addition to the encoded data itself, is better able to estimate the centroids during K-means iterations eventually leading to better clustering effectiveness in comparison to a range of baseline approaches for privacy preserving clustering. Further, the proposed PPK-means method (multi-component variant) is less computationally expensive than the standard K-means method. It was observed that on text data, the proposed method outperforms the standard K-means.

Chapter 6

Privacy Aware Semi-Supervised Learning [*]

In the previous chapter, we have explored unsupervised learning under the constraints of privacy preservation. In this chapter we will explore the privacy preserving semi-supervised learning to address our second research question RQ-2 introduced in the Chapter 1:

"*How the effectiveness of privacy preserving clustering on discrete metric space can be improved with weak supervision on the encoding transformation?*"

Chapter 6 is organized as follows. In Section 6.1, we introduce our contribution in the area of weakly supervised learning. Next, in Section 6.2, we present a novel weakly supervised deep metric learning framework towards privacy preserving clustering. Section 6.3 describes the experimental setup of the proposed approach. The results of our empirical evaluation studies have been presented in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.1 Introduction

In the present era of 'big-data' driven learning, Machine Learning as a Service (MLaaS) [169, 144] is a potential solution to meet the ever increasing computational requirements for training models requiring massive quantities of data. To avail MLaaS, a

[*] Some material from [20] has been reused in this chapter.

client first needs to upload the data, on which the computation is to be performed, to a server. While this workflow is suitable for a perfectly trusted server, in practice, however, no server is entirely trustable, because of the potential presence of malwares. This in turn may cause a breach in data privacy as the information in the original data may be used in undesirable ways [3, 144].

Generally speaking, existing research have demonstrated that an application of distance preserving, or locality sensitive encoding of the data often leads to improvements in unsupervised privacy-preserved downstream tasks, such as clustering [19, 54, 148], entity resolution [137, 72], etc. In particular, using a binary transformation as a particular choice for the encoding function has not only been shown to protect the integrity of the data [68], but has also been shown to provide computational and storage benefits [19]. Moreover, it has been shown that supplementing the encoded data with information on the global characteristics of the data distribution [19] produces effective clusters on the encoded space without compromising the privacy.

However, clustering on an encoded space of data instances obtained by standard encoding mechanism e.g. locality sensitive hashing can potentially be noisy [19]. We argue that clustering a small seed set of data on the client side and sharing the cluster membership information (not the original data instances themselves) with the server may potentially guide the clustering process on the server side. More concretely, in our proposed privacy preserving clustering workflow, we learn a distance metric function by leveraging the cluster membership information (hence, weak supervision) of the seed set of data to achieve effective encoding.

In contrast to the existing approaches of privacy preserving clustering, the works solely on a discrete metric space of encoded vectors, we propose a deep metric learning technique that uses the encoded space $\mathbb{H}^m$, Hamming space of dimension m, to train a transformation function with an objective to automatically learn the cluster affinity with respect to the original space $\mathbb{R}^d$, Euclidean space of dimension d.

Research objective. The objective of our present research in this chapter is to investigate ways of developing encoding functions that (in addition to maintaining data privacy) also ensures satisfactory clustering effectiveness on the encoded data.

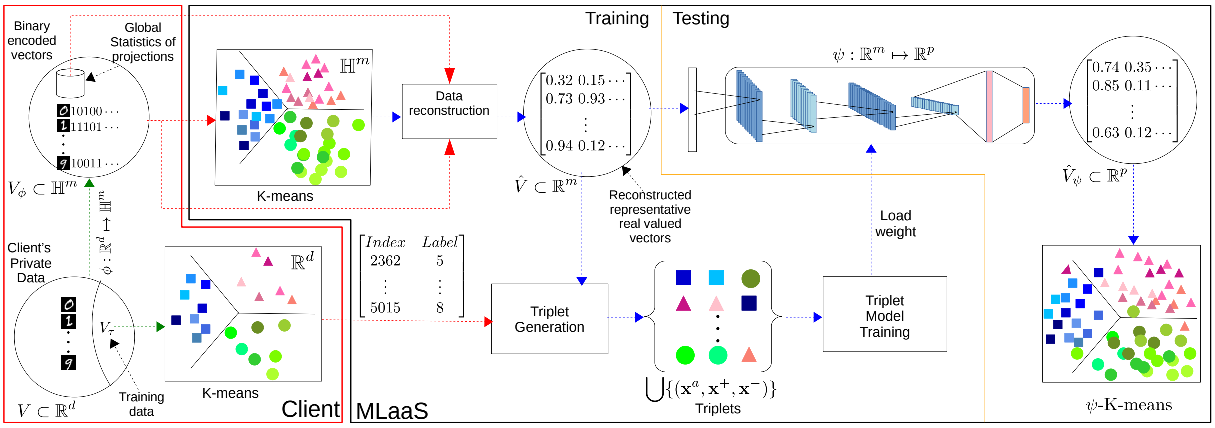


Figure 6.1: A schematic workflow of our proposed method. A detailed explanation of this figure is presented in Section 6.2.

Our Contributions. In summary, the overall contributions of the present work in this chapter are as follows.

- We show that a data-driven weakly supervised approach of learning the encoding function performs well for the clustering task in comparison to unsupervised approaches of encoding the data.

- We also show that the quantity of data required for this weak supervision is small, which makes it practically possible to execute clustering on a small subset of data at the client side itself. Sending this metadata information of cluster labels on a small seed set of data enables the server side to learn a parameterized distance function on the encoded data space, eventually leading to more effective clusters.

- We empirically demonstrate that an effective reconstruction can be achieved, in practice, if additional statistical information in the form of means and standard deviations of the projected values along each basis vector of the encoding transformation matrix is shared with the server. Sharing such global statistics about the data does not compromise the privacy. We have made available the source code for our experiments for reproducibility and further research purposes[2].

6.2 Proposed Approach

In this section, we describe the details of our proposed weakly supervised deep metric learning based framework that allows provision for effective clustering under privacy preservation constraints. A schematic workflow of our proposed method is presented in the Figure 6.1, which is to be interpreted as follows.

The circle along the bottom-left part of Figure 6.1 shows the data instances ($V = \{\mathbf{v} : \mathbf{v} \in \mathbb{R}^d\}$) that are to be clustered. The data encoded by the client (Section 6.2.1) is shown as the circle on the top-left part of the figure. The client now sends the entire data (shown as the top-left circle) along with the projected value statistics (Section 6.2.2) computed for each basis vector (shown as the cylinder) to the MLaaS (shown by the two red colored arrows).

Additionally, the client also sends the results of the clustering in the form of (index, cluster-id) pairs on a small seed set of points, V_τ, to the MLaaS, which is used to train the triplet network (Section 6.2.4). To cluster the encoded data instances ($\subset \mathbb{H}^m$) on the server side, they are first approximately reconstructed as real-valued vectors $\mathbb{R}^m$ (Section 6.2.3) and then the trained triplet network is used to apply the transformation $\psi : \mathbb{R}^m \mapsto \mathbb{R}^p$ (Section 6.2.4), after which K-means is employed on these transformed vectors (Section 6.2.5).

6.2.1 Binary Transformation of Data Instances

In Chapter 5 we have shown that the binary transformation or Hamming space transformation of the real-valued vectors into an anonymous space leads to privacy of the data being preserved. Following that observation, in the present work also, we employ the Super-Bit LSH algorithm as the Hamming space transformation function $\phi : \mathbf{w} \in \mathbb{R}^p \mapsto \mathbf{h} \in \mathbb{H}^m$, which is given by,

$$\mathbf{h}_i = \text{sgn}(\mathbf{w} \cdot \mathbf{b}_i), \;\; i = 1, \ldots, m, \tag{6.1}$$

where $\mathbf{b}_i$ represents the i^{th} basis vector among m randomly chosen basis vectors orthonormalizes by applying the Gram-Schmidt algorithm and $\text{sgn}(\cdot)$ is the sign func-

[2] https://github.com/chandanbiswas08/pp-clustering

tion that returns 0 for a negative value of the parameter, and 1 otherwise. Let the encoded set, obtained by applying the encoding function on each element of V, be V_ϕ, i.e.,

$$V_\phi = \{\mathbf{h} : \mathbf{h} = \phi(\mathbf{x}),\ \forall \mathbf{x} \in V\}. \tag{6.2}$$

Detail description about the Super-Bit LSH based binary transformation is given in the Section 3.2.

6.2.2 Clustering a Seed-set at the Client Side

To effectively reproduce the clustering behaviour of $\mathbb{R}^d$ on $\mathbb{H}^m$, as a first step, we select a small subset $V_\tau \subset V$ from the entire set of data instances. We eventually use this subset to *guide* the clustering process on the server side (more details in Section 6.2.4). The size of the subset is parameterized by $\tau \in [0, 1]$ such that $|V_\tau|/|V| = \tau$.

As the process for selecting V_τ is random, it is not possible to ensure the class distribution membership is consistent with that of the overall data. However, since the sampling is uniform it is also *expected* that the proportion of the class memberships is close to their proportion in the overall data. This could lead to some data instances of a specific class being more frequently occurring than others, which may affect the effectiveness of the triplet network.

However, it is not correct in this experiment setup to bias the sampling process so as to ensure a consistent class membership reflective of the one in the overall data. This is because that would mean that we make use of the class labels, which is something that we cannot assume to exist for clustering. The class labels are only to be used for the purpose of evaluation.

Now regarding the size of this subset for training the triplet network, keeping in mind that the objective of employing an MLaaS in the first place was to minimize the computational overhead of the client, the proportion τ is set to at most 0.1 (10%) for practical purposes. Despite the additional computational overhead, clustering a small subset of the data on the client side would allow provision for a more effective clustering on the rest of the data, as our experiments will demonstrate.

In our proposed workflow, the client employs K-means on V_τ to obtain a set of cluster ids or labels, i.e., $\forall \mathbf{x} \in V_\tau$, we obtain its cluster label, say $c_{(\mathbf{x})} \in \{1, \ldots, K\}$. We then share this set of cluster labels $C_\tau = \{c_{(\mathbf{x})} : \mathbf{x} \in V_\tau\}$ with the MLaaS along with the set of encoded data points V_ϕ (Equation 6.2). Figure 6.1 shows that in our implementation of the workflow, the client sends each cluster label along with an index (pointer) of the member of the set V_τ to which the label applies. It is important to note that sharing only the cluster labels does not lead to a breach of privacy at the server end because the shared data contains suggestive information about the relative proximity of the *encoded* data points only, and no information about the value of the data instances themselves.

6.2.3 Data Reconstruction

Before discussing the triplet network for learning a parameterized distance metric on the set of encoded data points, V_ϕ (Equation 6.2), we first describe how the aggregated statistics of the projected values along each of the m basis vectors can be used at the server's side to better train the triplet network (to be discussed in Section 6.2.4).

Similar to the PPK-means algorithm described in Chapter 5 (published in article [19]), along with the set of binary encoded vectors, we also send the means and the standard deviations of the projected values along each of the basis vectors. The set of means and the standard deviation values are represented as m dimensional vectors, denoted respectively as $\mathbf{\Phi}^\mu \in \mathbb{R}^m$ and $\mathbf{\Phi}^\sigma \in \mathbb{R}^m$, where

$$\mathbf{\Phi}_i^\mu = \frac{1}{|V|}\sum_{\mathbf{v}\in V} \mathbf{v} \cdot \mathbf{b}_i,\ \mathbf{\Phi}_i^\sigma = \sqrt{\frac{1}{|V|}\sum_{\mathbf{v}\in V}(\mathbf{v} \cdot \mathbf{b}_i - \mathbf{\Phi}^\mu{}_i)^2}. \tag{6.3}$$

The PPK-means approach of Chapter 5 have used the global projection statistics, i.e., the vectors $\mathbf{\Phi}^\mu$ and $\mathbf{\Phi}^\sigma$ to better estimate the centroid vectors while executing K-means on the binary encoded vectors. Instead of estimating the bits of a centroid vector, we use this additional information for an *approximate reconstruction* of the binary encoded data as real-valued vectors of the same dimension. In other words, we aim to reconstruct the projected values themselves from the signs of these values. The projected value along the i^{th} basis of data instance can be estimated by sampling

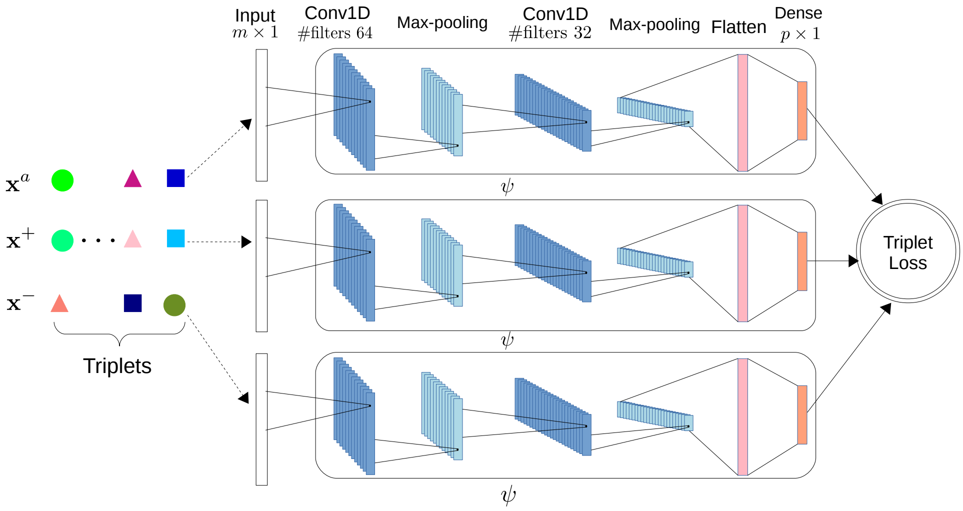


Figure 6.2: Architecture of the triplet network used in our privacy-preserved clustering workflow.

a value from the distribution $\mathcal{N}(\mathbf{\Phi}_i^\mu, \mathbf{\Phi}_i^\sigma)$ (sent by the client to the MLaaS as per Equation 6.3), on the basis of the assumption that these values are Normally distributed.

Additionally, to accommodate the priors of cluster affinities, we first apply K-means on the set of encoded vectors to obtain a cluster label, say $c_{(\mathbf{h})}$, for each $\mathbf{h} \in V_\phi$. For this step, the underlying space on which K-means operates is $\mathbb{R}^m (\supset \mathbb{H}^m \supset V_\phi)$ thus allowing a more accurate representation of the centroid vectors with non-binary components.

We then scale each sampled component $\mathbf{x}_i^a$ of the vector $\mathbf{x}^a$ with the prior likelihood of this cluster label $\forall i = 1, \ldots, m$. Formally speaking,

$$\mathbf{x}_i^a = \frac{|\{\mathbf{g} \in V_\phi : c_{(\mathbf{g})} = c_{(\mathbf{h})} \wedge \mathbf{g}_i = 1\}|}{|\{\mathbf{g} \in V_\phi : c_{(\mathbf{g})} = c_{(\mathbf{h})}\}|}\alpha_i, \quad \alpha_i \sim \mathcal{N}(\mathbf{\Phi}_i^\mu, \mathbf{\Phi}_i^\sigma). \tag{6.4}$$

In Equation 6.4, $\mathbf{h} \in \mathbb{H}^m$ denotes the binary vector that is to be reconstructed to an m-dimensional real-valued vector $\mathbf{x}^a$, α_i represents the posterior and is a random variable drawn from the global statistics of the projection values of the original data instances shared with the MLaaS (Equation 6.3).

Note that the data instance, $\mathbf{x}^a$, reconstructed from $\mathbf{h} \in \mathbb{H}^m$ is a vector embedded in $\mathbb{R}^m$. Subsequently, we train the metric learning network using the set of points $\mathbf{x}^a \in \mathbb{R}^m$ serving as anchor points for each triplet.

6.2.4 Metric Learning with Weak Supervision

We now discuss how we make use of the small seed set of cluster labels at the server side to better estimate the topology of the space. Recall from Section 6.2.2 that the clustering of the original data instances were conducted at the client side, and the only information revealed to the MLaaS about a data instance was it's cluster label. The idea here is to use these cluster labels as representative examples to learn transforming the encoded points in a way such that the distance between the points belonging to the same cluster are reduced and the those between different clusters are increased.

The standard way of achieving this is via a neural network with shared parameters [32], and employing a triplet loss to train the network. More concretely, each data instance to train such a network is a triplets of the form $(\mathbf{x}^a, \mathbf{x}^+, \mathbf{x}^-)$, where $\mathbf{x}^a$ is a pivot (anchor) point, $\mathbf{x}^+$ is a point that belongs to the same class (in our case, cluster), and $\mathbf{x}^-$ is a point from a different class [125], all these being reconstructed from the encoded data points V_ϕ (Equation 6.4).

After training a triplet network with triples of the form $(\mathbf{x}^a, \mathbf{x}^+, \mathbf{x}^-)$, during the testing phase only a part of the network corresponding to one input instance is used to obtain a vector of dimension p, the dimension of the last layer of the network. Mathematically speaking, given a test point (a vector of dimension m in our case) the trained network outputs a vector of dimension p, or in other words, we eventually learn a function $\psi : \mathbb{R}^m \mapsto \mathbb{R}^p$, ψ denoting the trained triplet network parameters. The loss function

$$J(\psi) = \min \sum_{(\mathbf{x}^a,\mathbf{x}^+,\mathbf{x}^-)} [|\psi(\mathbf{x}^a) - \psi(\mathbf{x}^+)| - |\psi(\mathbf{x}^a) - \psi(\mathbf{x}^-)| + M]_+ \tag{6.5}$$

is used to learn the parameterized distance function ψ aiming to minimize the distances between the points (in V_τ) observed to be in the same cluster and maximizing the distance between those observed to be in different clusters. In Equation 6.5, $\mathbf{x}^a \in \mathbb{R}^m$ is an approximate reconstruction of a data point from an encoded vector (Equation 6.4), $\mathbf{x}^+$ and $\mathbf{x}^-$ are the positive and negative examples for the anchor, M is the hinge-loss margin, and $[\cdot]_+$ is an abbreviation for the hinge loss function, $\max(0, \cdot)$.

As the input data points to the network, we use the approximately reconstructed data points $\mathbf{x}^a \in \mathbb{R}^m$ (Equation 6.4) instead of the encoded data points, $\mathbf{h} \in \mathbb{H}^m$, themselves. Moreover, in contrast to the standard approach of using true class labels to train triplet networks, (i.e. employing strong supervision) as in [125, 62, 60], we apply labels obtained in an unsupervised manner (in our case, by the application of clustering).

Figure 6.2 shows the triplet network that we employed. The cluster label information of the encoded points were used to form the triplets (each column of the left part of the figure), containing as examples a point from the same cluster and another from a different one (cluster label shown with a specific shape and a color, e.g. a green circle etc.). The architecture of the shared part of the network consists of two layers of 1D convolution operators (the kernel and the filter sizes employed in our experiments are shown in Figure 6.2).

The configuration of the network (number of layers, kernel sizes etc.) were adjusted with a grid search. We report the best configuration in Figure 6.2. The decision to apply one dimensional convolution (instead of higher dimensional convolution) as a standard feature extractor stems from the fact that the data instances themselves are 1^{st} order tensors (i.e., vectors), which in turn ensures that the same network can work for both text and images (a standard approach to convert a 2D image to a 1D vector is via a variational autoencoder (VAE) [76]).

6.2.5 Clustering at the Server Side

During the testing phase, the output, $\hat{V}_\psi = \{\mathbf{w} \in \mathbb{R}^p : \mathbf{w} = \psi(\mathbf{x}), \mathbf{x} \in \hat{V}\}$, from the penultimate layer of the network (discussed in Section 6.2.4) is used as the representation of an encoded data point. The advantage of this step is that the server side computation proceeds in a continuous metric space, $\mathbb{R}^p$, instead of a discrete metric space $\mathbb{H}^m$. It can be argued that this way of representing the data points in $\mathbb{R}^p$ better preserves the topology (relative neighborhoods) of the original data space $\mathbb{R}^d$ than the binary encoded space $\mathbb{H}^m$. As the final step of our method, we execute the K-means clustering algorithm on the embedded vectors $\hat{V}_\psi$ to yield the final clustering output.

6.3 Experimental Setup

We conduct a number of privacy-aware clustering experiments on various publicly available datasets to show the benefits of the proposed approach of employing a weak supervision based metric learning approach.

6.3.1 Datasets

We have conducted clustering experiments on four standard datasets of three different modalities, namely the MNIST [84], Fashion-MNIST [162], CIFAR-10 [79] and 20-Newsgroup (20NG) [82] datasets, first two datasets are gray image datasets, third one is color image dataset and the final one is a text dataset. For clustering evaluation, we consider the class labels of the respective datasets as the ground-truth. While the number of ground-truth clusters for all the three datasets MNIST, F-MNIST and CIFAR-10 is 10 and for the 20NG dataset this number is 20 (reflecting each topic). The detail description of the datasets are presented in the Section 4.1.

6.3.2 Model Training

Our proposed privacy-aware clustering workflow specifically the 1D CNN architecture based distance learning model was implemented in Keras. The model was trained with stochastic gradient descent. The value of the margin, M, of the triplet loss in Equation 6.5 was set to 0.2 as prescribed in [125].

6.3.3 Baselines

To compare the effectiveness of our proposed method, which we call ψ-K-means, we compare it with a number of different clustering approaches that work on binary encoded data. We also include the standard K-means executed on the real-valued Euclidean space as a reference for comparisons. Privacy is not preserved in this case, so instead of calling it as baseline we call it as *Apex*-line from which we can get idea about the best result which can be achieved with a ideal settings. particularly, the following privacy preserving clustering techniques are used as baselines:

1. **LSH-partition:** Locality Sensitive Hashing (LSH) is a popular data compression method where an identical hash code (called *signature*) is assigned to similar data vectors. A common LSH algorithm is *MinHash* that is expected to yield identical signatures for similar data instances using a permutation-based hashing scheme, as introduced in [64]. In the context of our experiments, we output a signature ranging from 1 to K for each vector, where K is the number of clusters. We then use the hash signatures for grouping together vectors with identical signature values. Since similar vectors are expected to produce same signatures, it is expected that similar points would cluster together.

2. **ϕ-K-means:** This baseline applies standard K-means algorithm on the binary encoded points, which are assumed to be embedded within the Euclidean space itself ($\mathbb{H}^m \subset \mathbb{R}^m$). Thus, the centroid vectors computed during the K-means iterations are vectors with real-valued components, instead of being binary. Note that this baseline has no way of adapting to the errors in the topology across the local neighborhood of points introduced during the encoding process.

3. **E2H-K-means:** This baseline, in contrast to ϕ-K-means, solely operates in the discrete Hamming space. The real-valued cluster centroids vectors computed during the K-means iterations are then truncated to the nearest point in the Hamming subspace ($\mathbb{H}^m \subset \mathbb{R}^m$) before assigning the cluster memberships to the points. It is likely that the distances in the discrete space are likely to exhibit a large number of ties, as a result of which the cluster membership assignments during K-means are also likely to be erroneous.

4. **PPK-means:** For this baseline, we apply the method described in Chapter 5 as well as the article [19]), which involves sharing of additional information in the form of global statistics of the projected values along each basis vector of the encoded space. This baseline method, although privacy-preserving, is unsupervised and has no means of adjusting the relative distances between the encoded points.

6.3.4 Ablations of ψ-K-means

We also experiment with the following ablations of our proposed method.

1. **H-ψ-K-means**: This ablation of ψ-K-means does not involve the data reconstruction step, i.e., it trains the triplet network on the m-dimensional binary vectors. The purpose of this ablation is to demonstrate the potential benefit of the data reconstruction step.

2. **ψ-K-means-NP**: This ablation of ψ-K-means, during the reconstruction step, does not scale each vector by the cluster membership priors in Equation 6.4, as a result of which the reconstruction of the vectors from binary to real-valued ones are carried out in a manner oblivious to the cluster memberships themselves. The suffix 'NP' in ψ-K-means-NP abbreviates 'no prior'.

6.3.5 Parameters and Evaluation Metrics

The parameter common to all the privacy-preserving clustering methods (i.e., the ones which operate on the binary encoded space) is the dimension m of the encoding space $\mathbb{H}^m$, or in other words, the number of basis vectors, into which the original d-dimensional real valued vectors are transformed at the client side. A parameter in the PPK-means baseline method is the number of components of a Gaussian mixture model which we set to 10 as prescribed in Chapter 5.

The parameters specific to ψ-K-means, H-ψ-K-meansand ψ-K-means-NP are following:

1. The dimension, p, of the embedding space $\mathbb{R}^p$ in which the binary vectors are eventually transformed, or in other words, the dimension of the last layer of the network in the Figure 6.2.

2. The proportion of data, τ, used as training data to generate the set of triplets for train the transformation function ψ.

We conducted a grid-search for the optimal values of m and p within the sets $\{32, 64, 128, 256\}$ and $\{64, 128, 256, 512\}$ respectively. Since the proportion of the seed set used to train the triplet network needs to be a small number (as the majority of the computation workload needs to be carried out at the server side), we conducted a grid search for optimal values of τ in $\{0.01, 0.02, 0.05, 0.1, 0.15\}$ (i.e. up to a maximum of 15% data was clustered at the client side).

For each method, we set the number of desired clusters of K-means (K's value) to the number of class labels in the corresponding dataset, i.e., 10 for MNIST, F-MNIST, CIFAR-10 datasets and 20 for 20NG dataset in all our experiments. We also conduct the experiments with varying number of desire clusters K within the sets $\{5, 10, 15, 20\}$ for the datasets MNIST, F-MNIST, CIFAR-10 and $\{10, 20, 30, 40\}$ for 20NG dataset to observe the effect of different number of desire clusters.

For measuring clustering effectiveness, we employ the standard pairwise accuracy based measures, namely F-score and Adjusted Rand Index (ARI) [151] (detail description of F-score and ARI is given in 4.4.1), whereas for measuring the homogeneity of the clusters we again employ a standard metric, namely the Normalized Mutual Information (NMI) [35](see 4.4.1 for description of NMI). Since our proposed workflow is a weakly supervised method, to enable fair comparisons with the other unsupervised baselines, we partition the set of instances, V, as $V_\tau \cup (V - V_\tau)$ for the unsupervised methods as well, after which we report the clustering effectiveness on the set $V - V_\tau$ only. The partitions for each method were chosen randomly with an identical seed value.

6.4 Results

In this section we present the results of our experiments. First, we compare the clustering effectiveness of our proposed method with other baselines. Then we analyze the efficiency (execution-time comparisons) of the clustering methods employed in our experiments. Finally, we conduct additional experiments investigating parameter sensitivity of the clustering methods.

Table 6.1: Results of comparative studies of ψ-K-means algorithm against baselines and ablation studies of the proposed strategy on MNIST ($K = 10$), F-MNIST ($K = 10$), CIFAR-10 ($K = 10$) and 20NG ($K = 20$) datasets. In each case, the best results among the privacy preserving approaches (excluding the standard K-means) have been bold-faced. Cells of this table that are not applicable to a method (such as the parameters m, p, τ for K-means) have been filled using gray color.

Dataset	Method Type	Method Name	Parameters					
			m	p	τ	F-score	ARI	NMI
MNIST	*Apex*-line	K-means				0.4295	0.3608	0.5181
	Baselines	LSH-partition	256			0.2373	0.0961	0.2354
		ϕ-K-means	128			0.3901	0.3209	0.4473
		E2H-K-means	256			0.3646	0.2912	0.4336
		PPK-means	256			0.4040	0.3361	0.4567
	Ablations	H-ψ-K-means	256	256	0.10	0.4174	0.3521	0.4533
		ψ-K-means-NP	256	512	0.10	0.3987	0.3309	0.4529
	Proposed	ψ-K-means	256	512	0.10	**0.4370**	**0.3711**	**0.4607**
F-MNIST	*Apex*-line	K-means				0.3805	0.3076	0.4706
	Baselines	LSH-partition	256			0.2177	0.0633	0.1985
		ϕ-K-means	256			0.4259	0.3584	0.5001
		E2H-K-means	256			0.3820	0.3084	0.4818
		PPK-means	256			0.4297	0.3627	0.5129
	Ablations	H-ψ-K-means	256	512	0.10	0.4348	0.3705	0.5077
		ψ-K-means-NP	256	512	0.10	0.4173	0.3479	0.5046
	Proposed	ψ-K-means	256	512	0.02	**0.4397**	**0.3730**	**0.5160**
CIFAR-10	*Apex*-line	K-means				0.1562	0.0607	0.1094
	Baselines	LSH-partition	256			0.1295	0.0035	0.0164
		ϕ-K-means	256			0.1362	0.0358	0.0761
		E2H-K-means	256			0.1311	0.0341	0.0622
		PPK-means	256			0.1465	0.0362	0.0752
	Ablations	H-ψ-K-means	256	512	0.10	0.1289	0.0316	0.0600
		ψ-K-means-NP	256	512	0.10	0.1481	0.0340	0.0720
	Proposed	ψ-K-means	256	512	0.02	**0.1513**	**0.0365**	**0.0769**
20NG	*Apex*-line	K-means				0.3240	0.2840	0.5014
	Baselines	LSH-partition	256			0.0968	0.0019	0.0679
		ϕ-K-means	256			0.3599	0.3232	0.5149
		E2H-K-means	256			0.3411	0.3048	0.4559
		PPK-means	256			0.3810	0.3458	0.5065
	Ablations	H-ψ-K-means	256	512	0.10	0.3364	0.2999	0.4702
		ψ-K-means-NP	256	512	0.10	0.3700	0.3352	0.5120
	Proposed	ψ-K-means	256	512	0.02	**0.3958**	**0.3630**	**0.5194**

Table 6.2: Comparisons of time requirements for execution of ψ-K-means algorithm versus baselines and ablation studies of the proposed strategy on MNIST ($K = 10$), F-MNIST ($K = 10$), CIFAR-10 ($K = 10$) and 20-NG ($K = 20$) datasets. Cells of this table that are not applicable to a method (such as the ϕ transformation time for K-means) have been filled using gray color. 6^{th} column of this table shows the time (in sec) required for each training epoch. "Data recon" represents the data reconstruction time for ψ-K-means-NP and ψ-K-means algorithms.

Dataset	Method Type	Method Name	Execution time (Sec)					
			ϕ	Data recon	ψ training	ψ	Clustering	Total
MNIST	*Apex*-line	K-means					3.88	3.88
	Baselines	LSH-partition	3.74				1.15	4.89
		ϕ-K-means	3.71				3.49	7.20
		E2H-K-means	3.69				15.02	18.71
		PPK-means	3.73				5.95	9.68
	Ablations	H-ψ-K-means	3.72		0.27	2.09	17.61	23.69
		ψ-K-means-NP	3.74	69.91	0.30	2.06	20.26	96.27
	Proposed	ψ-K-means	3.71	74.15	0.30	2.00	15.33	95.49
F-MNIST	*Apex*-line	K-means					3.95	3.95
	Baselines	LSH-partition	3.75				4.61	8.36
		ϕ-K-means	3.55				5.42	8.97
		E2H-K-means	3.52				14.51	18.03
		PPK-means	3.65				13.35	17.00
	Ablations	H-ψ-K-means	3.65		0.33	2.66	19.77	26.41
		ψ-K-means-NP	3.55	113.64	0.24	2.06	17.38	136.87
	Proposed	ψ-K-means	3.15	112.48	0.24	2.06	15.58	133.51
CIFAR-10	*Apex*-line	K-means					5.12	5.12
	Baselines	LSH-partition	3.28				4.01	7.29
		ϕ-K-means	3.27				4.95	8.22
		E2H-K-means	3.27				13.06	16.33
		PPK-means	3.27				9.55	12.82
	Ablations	H-ψ-K-means	3.28		0.35	1.91	22.29	27.83
		ψ-K-means-NP	3.27	124.57	0.48	2.11	21.52	151.95
	Proposed	ψ-K-means	3.27	125.80	0.49	1.86	20.66	152.08
20NG	*Apex*-line	K-means					3.48	3.48
	Baselines	LSH-partition	2.68				2.96	5.64
		ϕ-K-means	2.51				3.50	6.01
		E2H-K-means	2.63				15.77	18.40
		PPK-means	2.66				3.60	6.26
	Ablations	H-ψ-K-means	2.52		1.56	3.32	10.66	18.06
		ψ-K-means-NP	2.61	34.51	1.36	3.30	8.20	49.98
	Proposed	ψ-K-means	2.56	35.23	1.44	3.29	6.89	49.41

6.4.1 Comparison of the Clustering Methods

Table 6.1 presents the clustering results for the different methods investigated on the MNIST, F-MNIST, CIFAR-10 and 20NG datasets. The standard K-means algorithm without the privacy preservation constraint operates on the true (unencoded) data instances, as a result of which it yields effective results. Among the privacy-aware baselines, it turns out that for both image and text data PPK-means outperforms the other baselines.

The 'no prior' ablation method, i.e., ψ-K-means-NP, leads to clustering effectiveness close to ϕ-K-means, which shows that a simple reconstruction via sampling without using the prior likelihoods of cluster memberships does not lead to accurately reconstructing the data, as a result of which the benefits of employing the triplet network is not realized. The other ablation, H-ψ-K-means, which trains the triplet network on the encoded binary vectors, also does not perform the best mainly because of the discrete nature of the input (the zeroes in the input does not allow back-propagation via all paths of the network [135]).

The unsupervised clustering task on CIFAR-10 dataset turns out to be the most difficult task among the four datasets. This is likely because the dataset is comprised of color images of considerable complexity (involving multiple objects).

From Table 6.1 we can observe that standard K-means yields effective results on MNIST, F-MNIST and 20NG datasets, whereas for the CIFAR-10 dataset standard K-means yields a relatively poor result compared to other three. However, the important point to note is that the effectiveness of our proposed method, ψ-K-means, is still comparable with that of the standard K-means, and it still outperforms the other baselines even on the CIFAR-10 dataset.

The fact that our proposed approach, ψ-K-means, produces the best results demonstrates that, firstly, a (weakly) supervised approach that guides the clustering process is beneficial for encoded data, and that secondly, for the distance metric learning to work well, it is important to train the distance metric on real-valued inputs, which in the absence of the real data itself has to be reconstructed. It turns out that using the cluster membership priors (Equation 6.4) leads to an reasonably accurate reconstruction of the real-valued data instances, which eventually leads to better training the triplet network and hence an effective clustering of the encoded data.

In Table 6.2, we report a comparison of the execution time of our proposed method with respect to the other baselines. From the table, we observe that the execution time of our proposed method is the highest among the other methods investigated. However, we would like to mention that this is the trade-off which is to be accepted for obtaining better effectiveness. From a practical standpoint, the run-times of our proposed method are not markedly worse in comparison to the baselines and the ablations.

We provide some further analysis on the trade-off between effectiveness and efficiency by tuning the parameters m and p in the next section.

6.4.2 Parameter Sensitivity Analysis

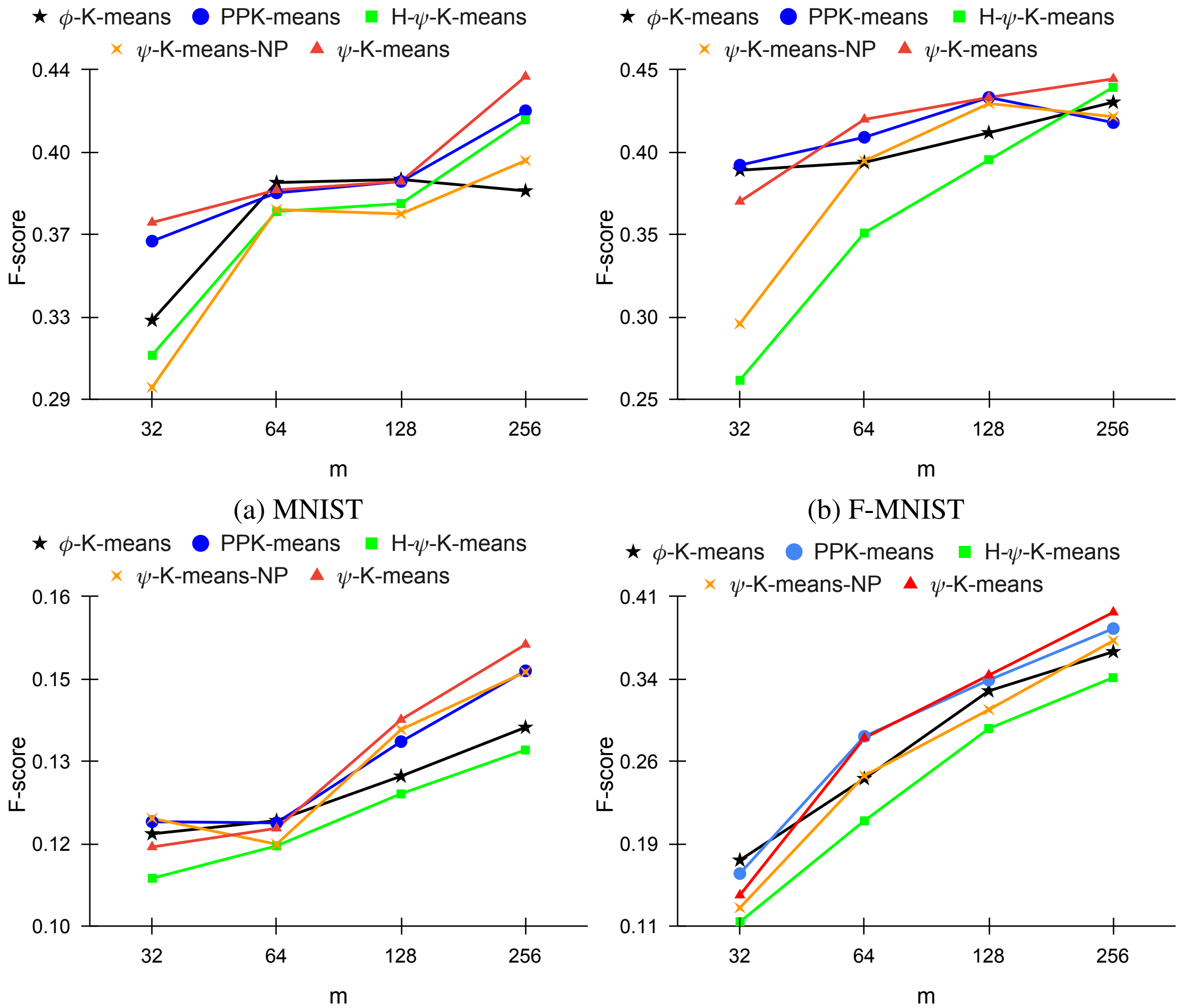
Figures 6.3 and 6.4 shows the sensitivity with respect to the dimension of the Hamming encoding space (m), and that with respect to the dimension of embedding space p, for the best performing clustering approaches. Moreover, Figure 6.5 analyzes the execution time of our proposed method for different values of m and p.

It can be observed, as expected, that increasing the dimension, m, of the Hamming space mostly improves the performance of both ϕ-K-means and ψ-K-means, because a higher dimension encapsulates more information about the topology of the original space.

Next, we investigate the effect of the parameter p (the output dimension of the triplet network) for our method and its ablation variants. Similar to the observation in Figure 6.3, Figure 6.4 also shows that increasing the dimension, p, of the metric learning transformation leads to better results.

Figure 6.5 shows that using higher dimensionality for m and p increases the execution time of our proposed method. We also observe in Figure 6.5 that the execution time of our proposed method increases at a higher rate with an increase in the parameter m than with respect to p.

Due to the increased computational complexity and memory requirements, it is



(a) MNIST

(b) F-MNIST

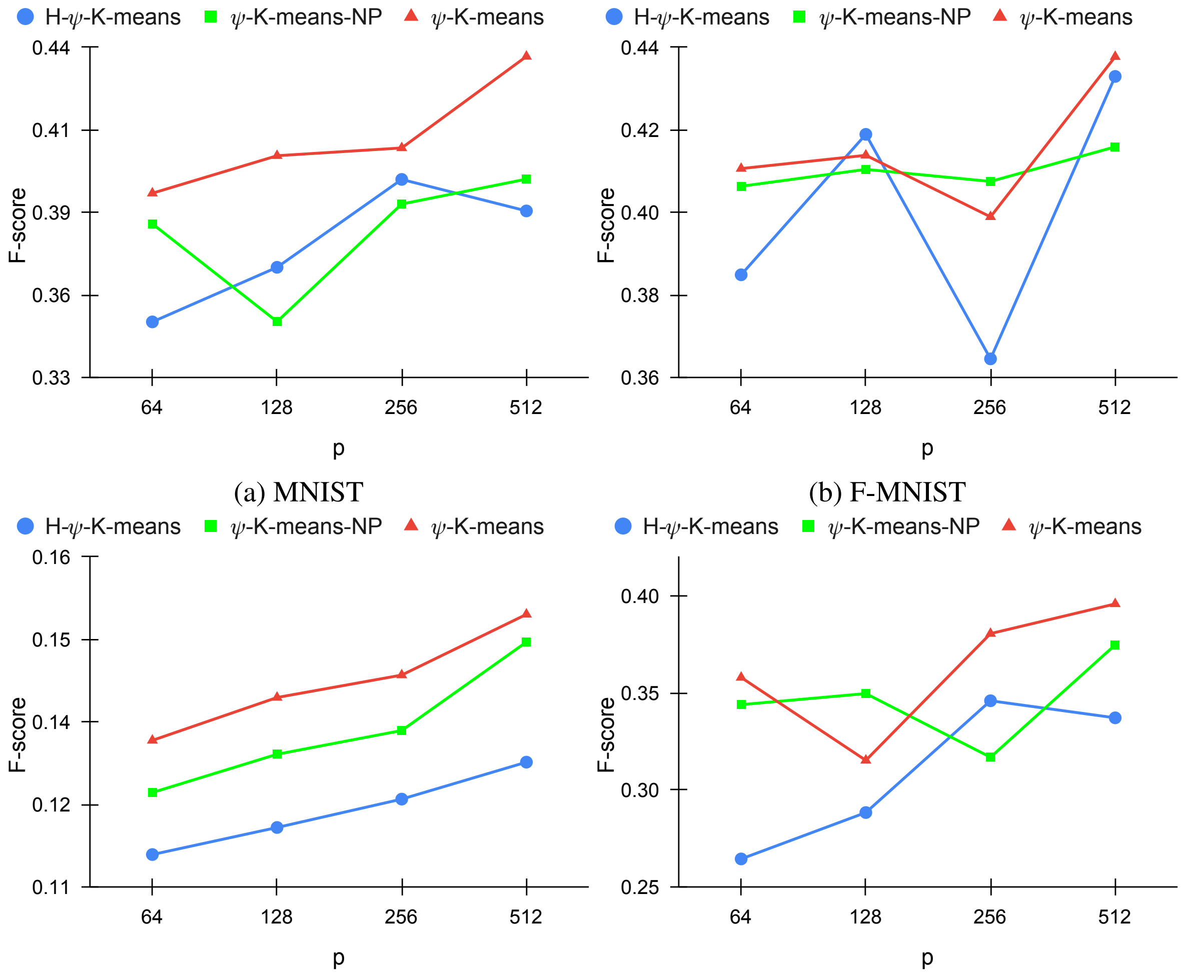
(c) CIFAR-10

(d) 20NG

Figure 6.3: Comparative performance of ϕ-K-means, PPK-means, H-ψ-K-means, ψ-K-means-NP and ψ-K-means clustering methods versus the Hamming space encoding dimension (m) corresponding to all the four datasets MNIST, F-MNIST, CIFAR-10 and 20NG. In each case, the comparison had been made by setting $p = 512$ (the optimal value) and the seed data proportion (τ) = 0.10.

however not practical to use substantially large values of m and p. Consequently, for achieving a trade-off between the execution time and the clustering effectiveness, we set the maximum values of m and p to 256 and 512, respectively, in our experiments. Moreover, we ensure that the value of m is not too large to reduce the data transmission overhead over the network.

The sensitivity of τ (the fraction of data used as training set) in ψ-K-means is presented in the Figure 6.6. We observe that even with parsimonious settings of using a small seed set, i.e., 1% of the data, we achieve results that are comparable to when 10% of data is used as the seed. Another interesting observation is that, increasing



(a) MNIST

(b) F-MNIST

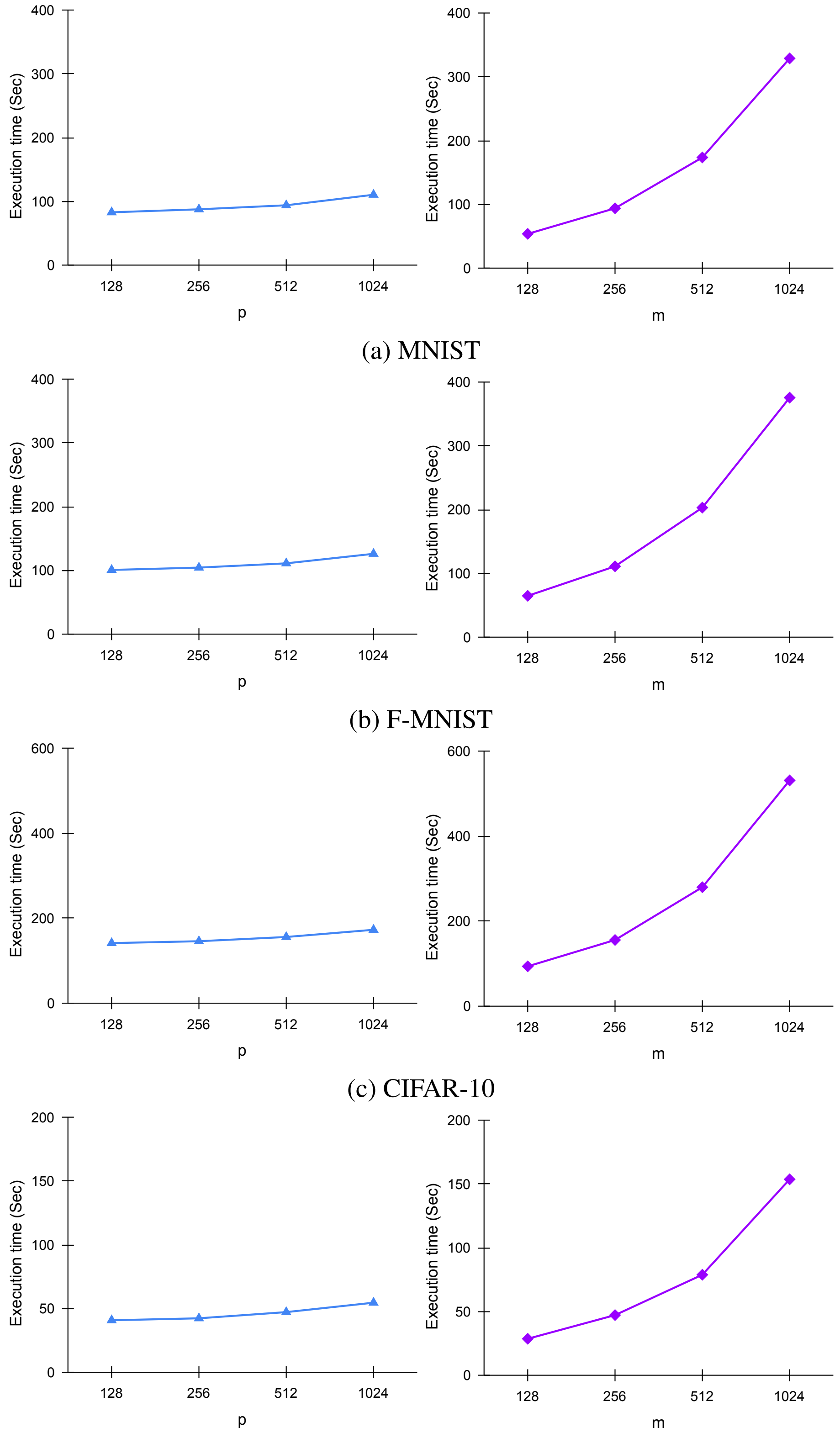
(c) CIFAR-10

(d) 20NG

Figure 6.4: Comparative performance of H-ψ-K-means, ψ-K-means-NP and ψ-K-means clustering methods versus the dimension of the target embedding space (p) of the triplet network corresponding to all the four datasets MNIST, F-MNIST, CIFAR-10 and 20NG. In each case, m, the Hamming space encoding dimension is set to the optimal value 512 and τ, the seed data proportion value to 0.10.

the value of τ the F-score value improves but we do not observe much improvement with the value of τ greater than 0.1. It is also to be noted that a higher value of τ leads to a larger volume of the seed set which increases the computational overhead in the client side. Therefore, the value of the fraction τ should preferably be close to 0.

Finally, Figure 6.7 shows the sensitivity of K (the number of desired clusters) on the clustering effectiveness. We can observe that our proposed method, ψ-K-means, yields the best results when the number of desired clusters, K, is set to the number of ground-truth classes for both the datasets MNIST and 20-NG. It is also seen that ψ-K-means consistently outperforms the baselines and the ablation methods for all values of K that we experimented with on both the datasets.



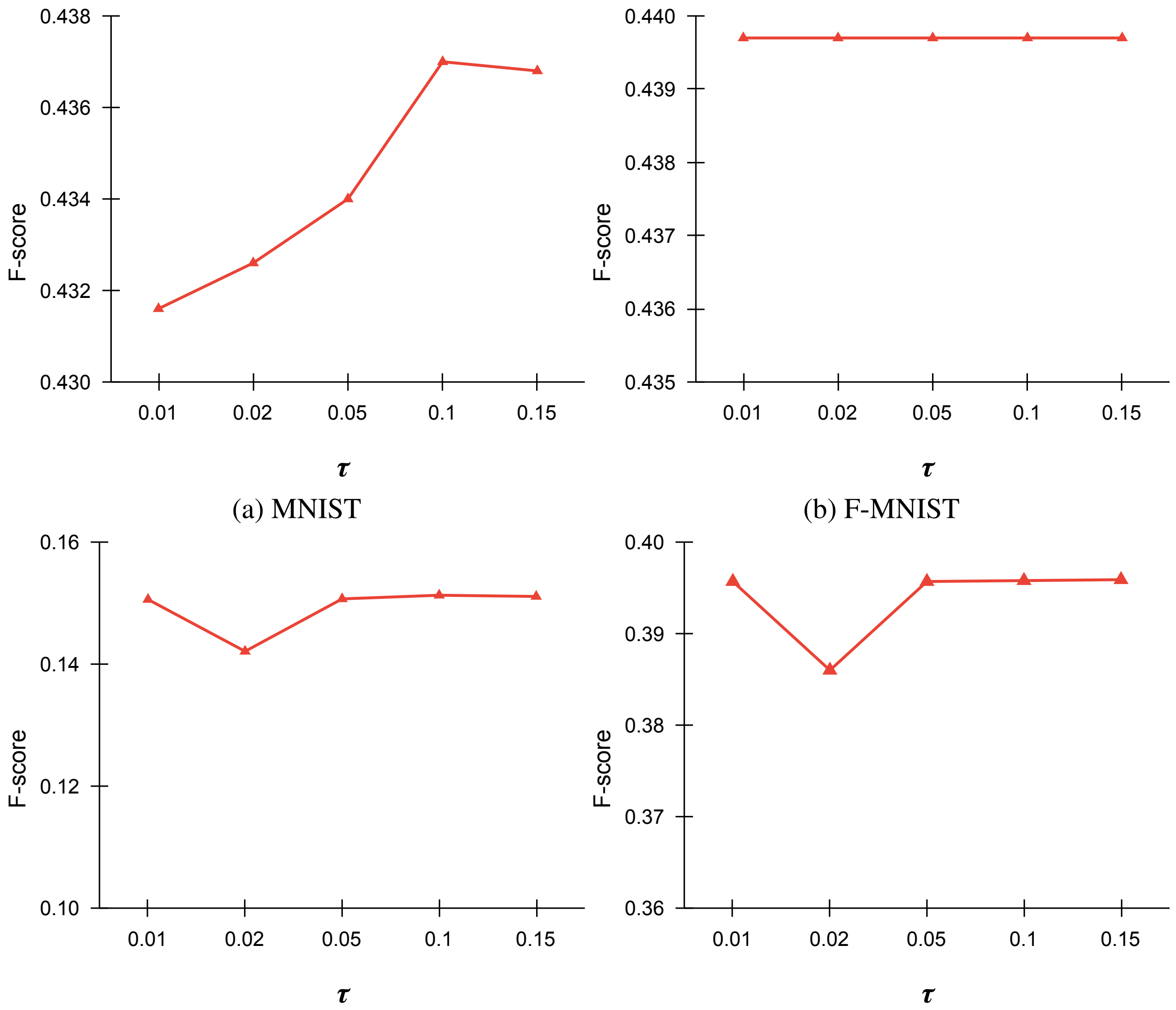
(a) MNIST

(b) F-MNIST

(c) CIFAR-10

(d) 20NG

Figure 6.5: Plot of execution time of ψ-K-means clustering method versus the Hamming space dimension (m) as well as the embedding space dimension (p) varying both between 128 to 1024 in multiples of 2, for the four different datasets MNIST, F-MNIST, CIFAR-10 and 20NG. The value of p is set to 512 while m is varied and the value of m is fixed at 256 while p is varied.



(a) MNIST

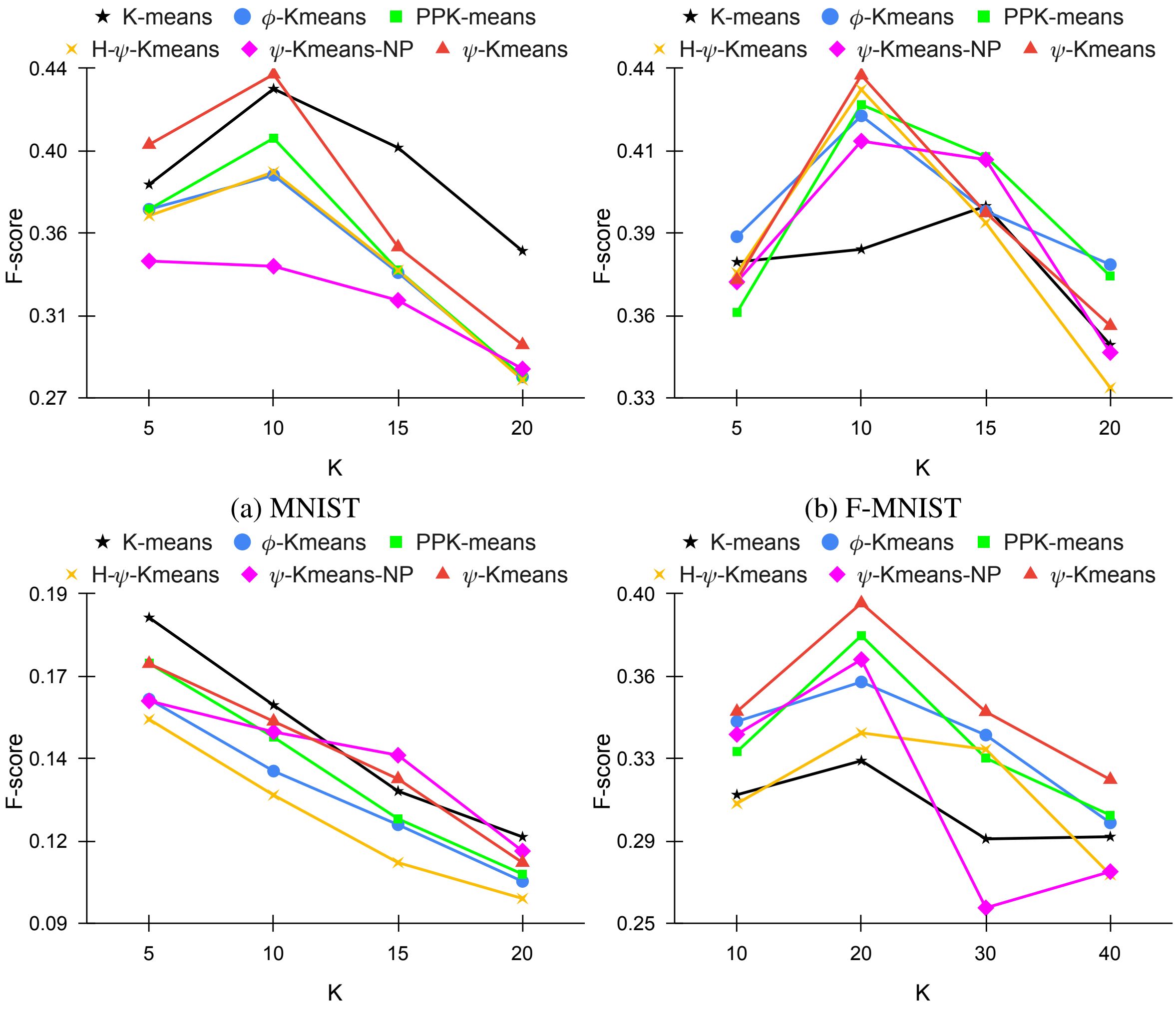
(b) F-MNIST

(c) CIFAR-10

(d) 20NG

Figure 6.6: Sensitivity of the value of τ, the fractional volume of the dataset used to train the triplet network towards clustering by ψ-K-means approach. Here, the values of the encoding dimension (m) and the target embedding space dimension (p) was set to the values 256 and 512 respectively for all the four datasets MNIST, F-MNIST, CIFAR-10 and 20NG.

The reason for this improvement over the baselines is that in ψ-K-means clustering operates on the representation of data points in $\mathbb{R}^p$, which, as we have already argued, caries better topology of the original data space $\mathbb{R}^d$ than the binary encoded space $\mathbb{H}^m$, on which the baselines ϕ-K-means and PPK-means operate. In addition, the reason why ψ-K-means outperforms the ablation methods is that the triplet network trained with reconstructed data using the prior likelihoods of cluster memberships potentially leads to better capturing the topology of the original data space in comparison to H-ψ-K-meansand ψ-K-means-NP.



(a) MNIST

(b) F-MNIST

(c) CIFAR-10

(d) 20NG

Figure 6.7: Sensitivity of the methods investigated with respect to the number of clusters. For our method, the reported results use the optimal value of m and p, i.e., 256 and 512 respectively for all the four datasets MNIST, F-MNIST, CIFAR-10 and 20NG. The value of τ, the proportion of seed data, was set to 0.10.

6.5 Conclusions

Privacy preservation has become an essential need in the present era of machine learning as a service (MLaaS). This chapter has particularly focused on the task of clustering massively voluminous data, for which a client may essentially need to use the computational resources of an external server. In similar scenarios, encoding of input data is crucial to ensure preservation of data privacy. Here, it has been hypothesized that effective controlling of the clustering process on encoded data should lead to useful clustering of the original data. Specifically, here we have proposed a metric learning based approach which consists of the reconstruction of a real-valued function approximating the data instances through the leveraging of additional statistics of the projected values along each basis vector used in the encoding process, and the training of a triplet network based on these reconstructed data instances using a small seed set of cluster membership associations. Here, it may be noted that the seed set has been clustered at the client's end while the membership information has been shared with the external server. The experiments presented in this Chapter have demonstrated that the proposed workflow is capable of producing improved clustering of the encoded data instances. Moreover, the ablation study performed by us has shown that an approximate reconstruction of real-valued data instances leads to better training of the triplet network, and subsequently the same leads to better effectiveness of the clustering process.

In the next chapter, we shall present a novel approach of privacy preserving supervised classification which successfuly implements defence against adversarial threats and the same will be preceded by a formal description of the general framework of privacy aware encoding.

Chapter 7

Privacy Aware Supervised Learning [*]

In Chapters 5 and 6, we have presented respectively an unsupervised and a semi-supervised learning algorithm capable of providing good performance under the privacy preservation constraints. In both cases, the privacy preservation has been achieved using a Hamming space transformation, in particular we have used Super-Bit Locality Sensitive Hashing as the transformation function. In this chapter, we shall focus on the supervised learning framework, in particular the supervised classification task, under the constraint of data privacy preservation to address our third research question RQ-3, which is

> *"How supervised learning can be used to defend the malicious attempts of stealing sensitive information from data shared on cloud platforms?"*

The rest of the chapter is organized as follows. In Section 7.1, we start with a brief introduction of privacy preserving supervised classification and our related contribution. In Section 7.2, we formally describe a general framework of privacy aware encoding, followed by our proposed model for defence against adversarial threats in Section 7.3. In Section 7.4, we present the experimental setup. Finally, Section 7.6 concludes this chapter.

[*] Some material from [18] has been reused in this chapter.

7.1 Introduction

The era of data-driven learning is continuously witnessing increased computational requirements for training multi-layered complex neural networks for supervised machine learning (ML) through a layered approach of abstraction from the raw data, e.g., the work on contextual word vectors pre-trained on large collections of documents to capture the inherent language model in text [36], or that of training deep image networks to capture higher levels of visual features from images [140].

One standard solution to mitigate the intensive computational requirements of training data-driven models is to follow the standard 'software as a service' paradigm, in which the computations to train an ML model are provided as a service (MLaaS) by a powerful computing device (server), virtually accessible through a distributed computing environment (cloud) [117]. An MLaaS-based solution requires a user (client program) to upload an encoded form of the data, usually corresponding to an abstract representation of it, e.g. pre-trained vectors such as BERT [36] for text, or Inception-Net for images [140]), to the server. Although such an MLaaS based workflow allows provision for distributed data sharing and also reduces the computational overhead of the client workstations, a risk with an MLaaS architecture is that it can potentially lead to breaches in data security and privacy [89].

To illustrate the point on potential threats on data privacy, consider an *adversarial model* which is able to eavesdrop on the communication channel between a client and the server offering computation on encoded forms of data. Imagine a situation where an adversarial model is *pre-trained* on past data, which in terms of its domain and characteristics, is similar to the one that is transmitted to the server over a communication channel. In such a situation, this pre-trained adversarial model could use this submitted data as an input to predict a number of *sensitive* attribute values from this data [34].

As a concrete example of an adversarial attack on data privacy, consider that the encoded data sent from a client workstation to a computation server over a communication channel corresponds to that of movie reviews, and the *primary task* for which the computational resources of the server is sought, refers to the task of classifying a

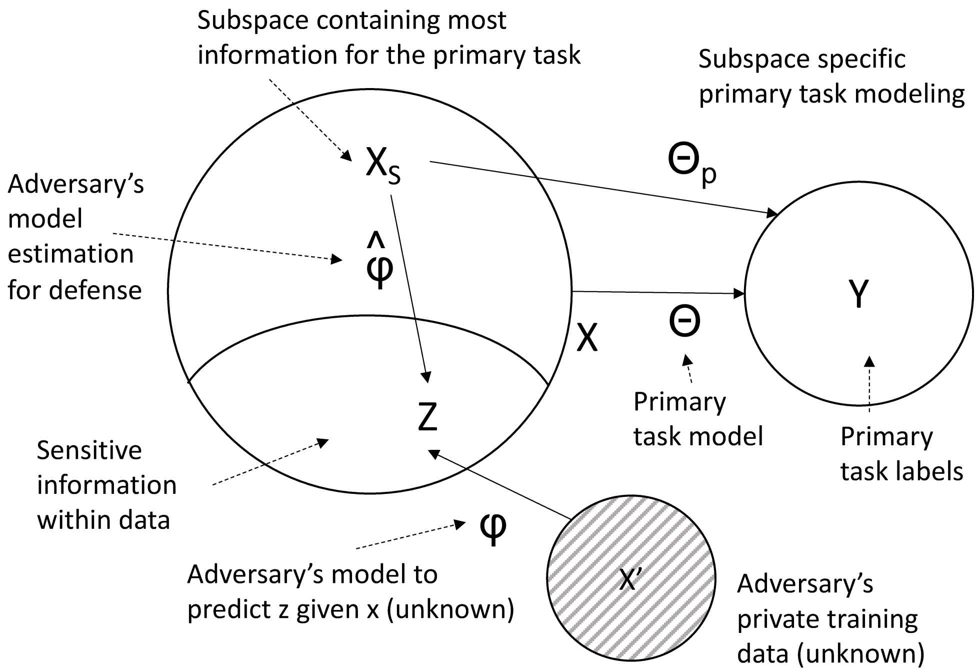

Subspace containing most information for the primary task
Subspace specific primary task modeling
Adversary's model estimation for defense
X_s
$\hat{\varphi}$
Θ_p
Θ
X
Y
Z
Primary task model
Primary task labels
Sensitive information within data
φ
Adversary's model to predict z given x (unknown)
X'
Adversary's private training data (unknown)

Figure 7.1: Schematic diagram depicting the proposed proposed defence mechanism against leakage of sensitive information; it works by identifying a candidate subspace, X_s, of the input space, on which the set of primary task labels, Y, is likely to exhibit some strong functional dependence. The remaining subspace, $X - X_s$, is used to estimate the possible functional dependence with the sensitive information, $\hat{\phi}$, an inversion on which is then used to defend against an adversarial model, ϕ.

review into positive or negative, i.e. the primary task involves learning a mapping of the form $\theta : \mathrm{x} \mapsto y$, $\mathrm{x} \in \mathbb{R}^d, y \in \{0, 1\}$, where x represents an encoding of the data, e.g. a sequence encoding of the words comprising the review [83]. Imagine that each review contains additional *identity information attributes*, z, corresponding to sensitive information about the author, e.g. the age, gender etc. Despite not being a part of the encoding, the adversary can potentially feed the encoded data as input into an adversarial network, that has already been trained on pairs of movie reviews encoding and the attribute values (e.g. gender), (x', z), to learn an association between the two of the form $\phi : \mathrm{x}' \mapsto z$, $\mathrm{x}' \in \mathbb{R}^d, z \in \{0, 1\}$. The parameters of the trained network, ϕ, may then accurately predict the demographics of the current encoded data x, i.e., the closer x is to x′ the higher is the associated risk of leaking the attribute value information [158].

A standard approach to prevent an attacker stealing the sensitive information from data is to make the encoding process itself aware of the intentions of an adversary, which usually involves first formulating the adversarial model, $\phi : \mathrm{x} \mapsto z$ as a secondary task, and then applying a multi-objective based encoding transformation of the data, where the first objective corresponds to the primary task and the subsequent ones correspond to one or more secondary tasks, each such secondary task representing an adversarial objective [34]. The learning objective, in this case, seeks to minimize the potential degradation of the primary task effectiveness due to the noise which is required to be incorporated within the data as a defence against adversarial attacks.

Our Contributions. We now enlist our contributions in this chapter. First, contrary to a standard approach of data-driven encoding that uses uniform weights for the abstract features, we hypothesize that the defence mechanism of a multi-objective based approach can potentially be improved by a weighted distribution over features. Specifically, this involves leveraging information from *candidate subspaces*, $\mathbf{x_s} \in \mathbb{R}^k, (k < d)$ of the input data that are *strongly correlated with the primary category labels* in the form $\theta_p : \mathbf{x_s} \mapsto y$. The *residual subspace* is thus likely to be functionally associated to the latent attribute values of the data, or in other words, to the secondary (adversarial) task categories $\hat{\phi} : \mathbf{x_s}' \mapsto z, \mathbf{x_s}' \in \mathbb{R}^{d-k}$, which in turn approximately models the function $\phi : \mathbf{x} \mapsto z$. We argue that this way of modeling the adversarial information yields a more robust encoding mechanism that is likely to be more resilient to security threats and our experiments confirm this hypothesis.

Second, in contrast to most existing approaches which conduct experiments mostly on text data with annotated metadata information (such as the demographic attributes, e.g., age and gender annotated as a part of the TrustPilot dataset [34]), we report empirical results on both images and text. For images, we test our method both on implicit and explicit demographic attributes. As implicit attributes, we use stylistic attributes, such as the slant or ligatures in handwriting, that could potentially reveal the age of a person. As explicit attributes, we test if the metadata information of age and gender associated with a set of lesion images can potentially be revealed to information stealing attacks.

7.2 A General Framework for Privacy-Aware Encoding

In this section, we formally describe a general framework for defence against adversarial threats using a multi-task learning based workflow. We present a general approach to the problem in the sense that the overall framework allows provision to incorporate more than one adversarial task, each corresponding to a particular attribute of the data.

7.2.1 Privacy-Agnostic Encoding

Using the notations introduced Section 7.1, the predictive model for the primary task, generally speaking, can be *learned* with a set of linear transformation functions (realized with a multi-layer perceptron) of the form

$$P(y=i|\mathbf{w};\theta,\theta_p) = \sigma(\theta_p \cdot \mathbf{x})_i = \frac{\exp(\theta_{p_i} \cdot \theta \cdot \mathbf{w})}{\sum_{j=1}^{c} \exp(\theta_{p_j} \cdot \theta \cdot \mathbf{w})}, \mathbf{x} = \theta \cdot \mathbf{w},\ \ \mathbf{x} \in \mathbb{R}^s, \mathbf{w} \in \mathbb{R}^d,\ y \in \mathbb{Z}_c, \tag{7.1}$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes a d-dimensional vector representation (encoding) of the input data, $y \in \mathbb{Z}_c$ denotes a class label (one of c possible values) corresponding to the classification task, $\theta \in \mathbb{R}^{s \times d}$ denotes a matrix of parameters (a latent layer of a neural network), and $\theta_p \in \mathbb{R}^{c \times s}$ denotes a matrix of parameters specifically corresponding to the classification task ($\theta_{p_i} \in \mathbb{R}^s$ is the parameter vector for the i-th class). As a simplification, we do not explicitly include the bias parameter as a part of the softmax equations. Since the encoding process of Equation 7.1 does not explicitly take account an adversarial threat against a subset of data attributes, the encoding $\mathbf{x} \in \mathbb{R}^s$ is privacy-agnostic.

7.2.2 Privacy-Aware Encoding

An encoding space different from Equation 7.1 that explicitly addresses a set of sensitive attributes has been shown to be effective in defence against adversarial models [34]. However, the work in [34] addresses the defence mechanism for a single attribute only. Instead, we present a more general setup involving more than one attribute.

In the context of our work, the attributes manifest themselves as an implicit part of the data, or otherwise, it is straight-forward to remove the attributes before encoding the data [45]. In particular, we assume that the encoding of an input data instance, w, is a function of both the raw data itself, (say w) and its latent characteristics (sensitive attributes). We represent a pair comprising an input data instance and a set of M sensitive attributes (assuming categorical values) associated with it as $(w, \{z_1, \ldots, z_M\})$, where $z_m \in \mathbb{Z}_{s_m}$, i.e. there are a total of s_j number of possible values for the j^{th} attribute.

A multi-objective transformation then uses the pairs, $(w, \{z_1, \ldots, z_M\})$, to encode the privacy-agnostic representation $\mathbf{w} \in \mathbb{R}^d$ as learnable parameters, $\mathbf{x} \in \mathbb{R}^s$, with the combined objective

$$P(y = i, z_1, \ldots, z_M | \mathbf{w}; \theta, \theta_p, \phi^1, \ldots, \phi^M) = (1 - \sum_{m=1}^{M} \gamma_m) \sigma(\theta_p \cdot \mathbf{x})_i - \sum_{m=1}^{M} \gamma_m \sigma(\phi^m \cdot \mathbf{x})_{z_m}, \quad (7.2)$$

where $\mathbf{x} = \theta \cdot \mathbf{w}$, $\mathbf{x} \in \mathbb{R}^s$, and $\mathbf{w} \in \mathbb{R}^d$, and similar to Equation 7.1, $\sigma(.)_i$ is an abbreviation for the softmax function with respect to the i-th class. The multi-objective loss of Equation 7.2 can be realized with a feed-forward network comprising a shared layer (parameter matrix $\theta \in \mathbb{R}^{s \times d}$) and the task specific layers. Separate layers, one for each adversarial task ($\phi^m \in \mathbb{R}^{s_m \times s}$), in addition to the primary task itself ($\theta_p \in \mathbb{R}^{c \times s}$), are all connected to the shared layer. Note that the parameters corresponding to w's in Equation 7.2 are obtained from pre-trained representations and hence are not learnable.

To illustrate Equation 7.2 with an example, consider a text classification problem, where each document is associated with the demographic attributes - age (z_1) and gender (z_2) of author. In such a situation, the value of M in Equation 7.2 would be 2. Continuing with the example, if age is discretized into 3 categories, e.g., 'young', 'middle-aged' and 'senior' then $s_1 = 3$.

In a generalized setting, the multi-objective loss function of Equation 7.2 models a relative trade-off between the effectiveness of the primary task and the desired lack of effectiveness of the adversarial ones (notice the negative factor in the linear combination corresponding to the adversarial tasks). A low value of each linear combination parameter, $\gamma_m \in [0, 1] : (\sum_m \gamma_m < 1)$, associates a small importance to the necessity of defending against an information stealing attack against the m-th attribute. Notice that setting $\gamma_m = 0$ degenerates Equation 7.2 to the privacy-agnostic encoding of Equation 7.1.

7.3 An Information Theoretic Perspective

In this section, we describe how to extend the general multi-task based privacy preserving approach from an information theoretic perspective. As per the motivation behind the schematic depiction of Figure 7.1, we now formally describe how to leverage information from the importance of features (components of the encoded vector representation of a data instance) to help the process of learning a better encoding for privacy preservation.

7.3.1 Subspace Encoding

A limitation of Equation 7.2 is that the parameters of the shared layer and the primary-task specific layer (i.e. θ and θ_p respectively) are trained with respect to the entire feature space of the encoded vector $\mathbf{w}$, whereas it is more likely to be the case that a part of this feature space correlates strongly with the primary task. The key idea in our proposed method is to substitute the encoding $\mathbf{w}$ of Equation 7.2 with a subset of features that are most likely to be informative for the primary task. This has a two-fold advantage.

First, a subspace of the most informative features for the primary task is likely to lead to a down-weighting of the residual subspace potentially constituting information responsible for determining the values of the sensitive attributes of the data. In other words, this is likely to degrade the effectiveness of the secondary tasks thus providing a potentially improved defence mechanism.

Second, since the subspace-based encoding approach puts more emphasis on parts of the data that are potentially responsible for determining the primary task output, it is also likely to lead to improving the effectiveness of the primary task itself.

7.3.2 Parameterized Subspace Selection with Gumbel Distribution

The authors of [30] computed the importance of features by measuring the mutual information between the primary task labels and an arbitrary feature subspace $\mathbf{w_s} \in \mathbb{R}^k, (k < d)$. The total number of possible subspaces, $\binom{d}{k}$, is exponential for relatively large values of k. Hence finding an optimal subspace representing the largest amount of information for data driven models is a challenging problem. A solution, proposed in [30, 67], is to use a parameterized version of a subspace (specifically obtained with a Gumbel distribution) that allows a gradient descent based optimization of its parameters. The objective is seek an optimum state of maximum informativeness of the subspace with respect to a set of labels. Before describing how this is applied in the context of our problem, we present a brief overview of the Gumbel based learning of subspaces, mostly following the exposition of [30].

A Gumbel distribution, $G(0, 1)$, is a distribution of random variables of the form $G_i = -\log(-\log u_i), u_i \sim \mathcal{U}(0, 1), \mathcal{U}$ being the uniform distribution. The Gumbel softmax probability distribution uses a concrete distribution, which is a continuous differentiable approximation of a categorical random variable. The *Gumbel softmax* is a modification of the softmax function involving random variables sampled from the Gumbel distribution, one each for each component of the softmax. In the context of our problem, we use the Gumbel softmax distribution to estimate the importance of each component of the encoding vector, $\mathbf{w} \in \mathbb{R}^d$. Formally speaking,

$$C = \{C_i : C_i = \frac{\exp((\log w_i + G_i)/\rho)}{\sum_{j=1}^{d} \exp((\log w_j + G_j)/\rho)}, \quad i = 1, \ldots, d\}, \tag{7.3}$$

where ρ is a *temperature* parameter, higher values of which makes the distribution close to uniform (for our experiments, we set $\rho = 0.1$ as per [30]). To select k features from a set of available d features, one needs to independently sample k times from the Gumbel softmax distribution resulting in a total of k random vectors $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$, where the j^{th} vector $\mathbf{c}_j$ is sampled from Gumbel softmax, i.e., $\mathbf{c}_j \sim C$. Let $\Lambda_k \in \mathbb{R}^{d \times k}$ be the matrix constituted from the k random vectors, $\mathbf{c}_j$, thus sampled. A row-wise maximum of the matrix, Λ_k then yields an approximation of a k-hot random vector $\lambda_k \in \mathbb{R}^d$. The highest k elements of λ_k (corresponding to the most important features) are retained while the rest $(d-k)$ are set to 0. Thus λ_k is a vector with k non-zero elements (*soft* k-*hot*) determining the choice of a k-dimensional subspace.

7.3.3 Feature Subspace with Multi-Objective

In the context of our problem (see Equation 7.2), data is represented as vectors in d dimensions, i.e. $\mathbf{w} \in \mathbb{R}^d$, out of which we intend to select a subspace $\mathbf{w_s} \in \mathbb{R}^k$ comprised of the most informative features. After selecting a random vector with k non-zero elements, λ_k, we now model its interaction with the primary classification task as

$$P(y=i, z_1, \ldots, z_M | \mathbf{w}; \theta, \theta_p, \phi^1, \ldots, \phi^M) = (1 - \sum_{m=1}^{M} \gamma_m) \sigma(\theta_p \cdot \mathbf{x})_i - \sum_{m=1}^{M} \gamma_m \sigma(\phi^m \cdot \mathbf{x})_{z_m}, \quad (7.4)$$

where $\mathbf{x} = \theta \cdot (\mathbf{w} \odot \lambda_k)$, $\mathbf{x} \in \mathbb{R}^s$ and $\mathbf{w} \in \mathbb{R}^d$. Equation 7.4 is a more constrained form of Equation 7.2. This is because instead of considering an arbitrary s-dimensional transformation from $\mathbf{w}$ (privacy-agnostic encoding) to $\mathbf{x}$ (privacy-aware encoding) of Equation 7.2, we specifically select an informative subspace, denoted by, say $\mathbf{w_s} = \mathbf{w} \odot \lambda_k$. This is obtained by an element-wise multiplication of the input encoding with a soft k-hot vector obtained from the Gumbel softmax distribution.

As a next step, the informative subpace is used to learn the privacy-aware encoded representation. In our experiments, instead of specifying the value of k directly, we control it with a fraction, $\tau \in [0,1]$ of the input data dimension, i.e., $k = \lfloor \tau d \rfloor$.

7.4 Experimental Setup

7.4.1 Experiment Workflow

A laboratory based setup[2] is devoid of the presence of a true adversary (e.g. as shown in the schematic of Figure 7.1). In such a situation, the adversary would have access to a pre-trained model which is trained to predict the sensitive attributes from input data instances. An adversarial model is likely to be more harmful if it has been trained

[2] A prototype of the implementation is available at https://github.com/chandanbiswas08/l2x-mt

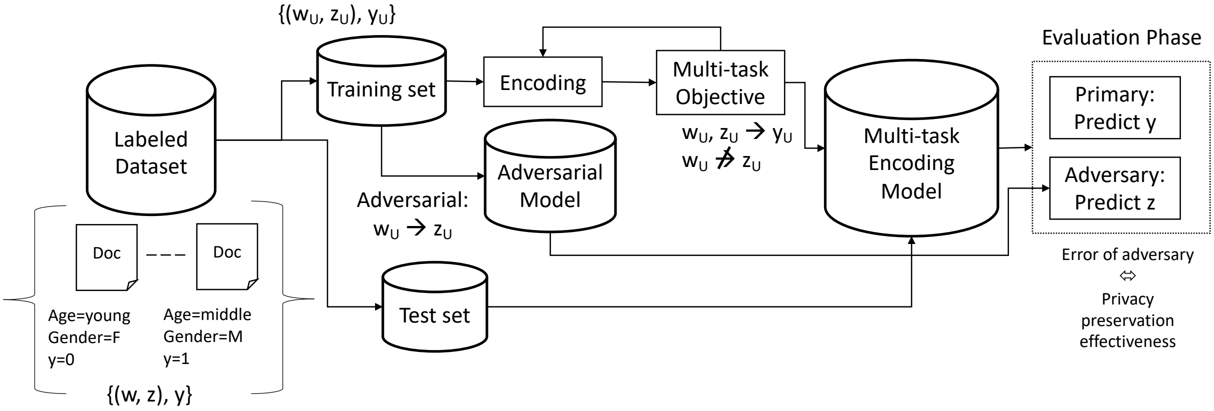


Figure 7.2: Schematics of the common setup for the evaluation workflow. Both the privacy-aware encoding and the adversarial model (one for each attribute) is trained on the training set of the data. During evaluation phase, the *privacy-preserved* encoded vectors for the test set are fed into the adversarial model to predict values of the attributes. The prediction error of this pseudo-adversarial setup indicates the effectiveness of privacy preservation.

on data instances that resemble the ones (i.e. similar in terms of encoded vector representations) to the ones that are sent over from the client to the MLaaS. To mimic this situation as closely as possible in a laboratory setup, we set up our experiments as shown in Figure 7.2.

For each labeled dataset, each data instance is annotated with additional attribute value pairs. With this we train a logistic regression model on the training set of the data to simulate an adversarial attack of predicting these additional attribute values from the data (a separate adversarial model is trained for each attribute type, shown as a single model in Figure 7.2 to avoid clutter).

In general, corresponding to M different attribute types (see Equations 7.2 and 7.4), we evaluate the effectiveness of the adversarial task as an *inverse effectiveness* measure for a particular defence method used in our experiments. The experiment workflow ensures that the encoding process of a defence mechanism is oblivious of the category values (e.g., values of age and gender) of the test-split.

7.4.2 Dataset

To test the effectiveness of our proposed subspace based privacy preservation approach on different modalities of data, we experiment with both text and image datasets, namely Skin Cancer MNIST (HAM10K) [146], Morpho-MNIST (M-MNIST) [24] and

TrustPilot [63], where first two datasets are colour image and gray scale image dataset respectively and the last dataset consists of text samples. The detail of these datasets are described in the Section 4.2.

7.4.3 Baselines

As baselines, we compare the following approaches. First, we apply a privacy agnostic logistic regression based approach (see Equation 7.1), which we denote as **LR**. Our next baseline, denoted as **MT**, is the multi-tasking based approach from existing literature [34], which we presented in this chapter as Equation 7.2. To explore if subspace based information usage, which forms a part of our proposed method, is indeed effective, we conduct experiments with two ablation baselines. The first of these baselines (applicable for text) involves the following. After computing the term feature weights with a simple term importance statistics (specifically tf-idf), for each sentence we retain only a fraction, $\tau \in [0, 1]$, of the terms with the highest weights. The rationale of this baseline, denoted as **LR-TFIDF**, is to see if removing a subset of features, not correlated to the primary task alone, can prevent information leakage of secondary attributes.

The second ablation baseline is a degenerate case of Equation 7.4, where we set $\gamma_m = 0$ for each adversarial task. This means that the k-dimensional encoding of the data, being agnostic of the adversarial tasks, only takes into account the informative subspace of the primary task. Unlike LR-TFIDF, this baseline method, denoted as **L2X** in our experiments, is applied to both text and images.

7.4.4 Evaluation Metrics and Parameters

As an evaluation metric, we employ a combination of the primary task accuracy (higher the better) and the inverse accuracy of the secondary tasks (lower the better) (The mathematical expression for accuracy is given in Equation 4.11). A high value of the combined metric reflects a better defence against information leakage without a substantial drop in primary task effectiveness. For combination, we specifically use the harmonic mean between the inverse of the aggregated accuracy values of the secondary tasks (A_S) and the accuracy of the primary task (A_P), i.e.,

$$F_{S_i} = \frac{2A_P(1 - A_{S_i})}{(1 - A_{S_i}) + A_P}, \tag{7.5}$$

and

$$F_S = \frac{2A_P(1 - A_S)}{(1 - A_S) + A_P}, \tag{7.6}$$

where A_S is the harmonic mean over the accuracy of each adversarial task, A_{S_i}.

The hyper-parameters tuned for each method were:

a) τ, which controls the number of features retained (for the LR-TFIDF baseline, this refers to the fraction of the terms retained with the highest tf-idf scores),

b) (γ_1, γ_2), which controls the relative importance of the two adversarial tasks (Equation 7.4).

In particular, the range of these hyper-parameters in our experiments were: $[0.2, 0.8]$ for τ, and $[0.1, 0.4]$ for γ_1 and γ_2, in steps of 0.2 and 0.1 respectively.

7.5 Results

7.5.1 Summarization of the Results

Table 7.1 summarizes the results of our experimentation of various privacy preserving supervised learning methods on different datasets. These results have been obtained after tuning hyper-parameter values of individual learning algorithms on the validation split of respective datasets.

It may be observed that although LR, being a privacy agnostic approach, provides high accuracies for the primary classification task, it also yields high accuracy for the adversarial tasks indicating a substantial leakage of information by the LR method. Multi-tasking based encoding (MT) strategy helps to improve the results, particularly for text samples, as it was also noted earlier in [34].

Subspace encoding alone (L2X) is also able to decrease the accuracy for the adversarial tasks (i.e. improve privacy preservation), which also means that a combination of MT and L2X should also improve results. This is precisely what is demonstrated by the results of our method (L2X-MT), which yields the best results for each dataset.

Here, we have used the McNemar hypothesis test [103] to determine whether the proportion of samples correctly classified in one scenario (say, using the LR method) is similar to the same of another scenario (say, using the proposed LR-MT method). Thus, our null hypothesis (H_0) is the classification performances in both the scenarios are identical. In order to realize the McNemar test, available results of our related experimentations have been arranged as in the following 2×2 contingency table, with the cell frequencies (A, B, C, D) equaling the respective numbers of pairs of the counts of individual samples undergoing classifications in the two individual scenarios.

		One Scenario	
		Correct	Wrong
Another Scenario	Correct	A	B
	Wrong	C	D

Cell values B and C had been used to compute the McNemar test statistic (Chi-Square) as follows:

$$\chi^2 = \frac{(B-C)^2}{B+C}$$

The above follows Chi-Square distribution with 1 d.f. and the testing has been conducted at 5% level of significance. Now, the p-value is the probability of observing this χ^2 value, assuming that the null hypothesis is true, and the two-sided p-value can be computed by:

$$\mathrm{p} = 2\sum_{i=B}^{n}\binom{n}{i}0.5^i(1-0.5)^{n-i} \tag{7.7}$$

where, $n = B + C$. Further details of this hypothesis testing have been provided in Section 4.4.2.

Table 7.2 presents the details of the results of the above hypothesis testing for comparing the proposed method (L2X-MT) with the baseline methods. Here, it may be observed that the p-value for comparison of L2X-MT method with the baseline LR method corresponding to the primary task is larger than the value 0.05 on each of the three datasets used in our study. Thus, the underlying null hypothesis can not be

Table 7.1: Summary of the results of experimentation of various baseline privacy preservation approaches along with our proposed method (L2X-MT) on different datasets. Cells of this table that are not applicable to a method (such as the parameters τ, γ_1 and γ_2 for method LR) have been filled using gray color.

		Hyper-parameters			Accuracy			Combined Measures		
Dataset	Method	τ	γ_1	γ_2	A_P	A_{S_1}	A_{S_2}	F_{S_1}	F_{S_2}	F_S
TrustPilot	LR				0.8674	0.7292	0.7168	0.4127	0.4270	0.4200
	LR-TFIDF	0.2			0.8194	0.7113	0.6928	0.4270	0.4469	0.4371
	MT		0.4	0.4	0.8694	0.6849	0.6920	0.4626	0.4549	0.4587
	L2X	0.2			**0.8726**	0.6804	0.6546	0.4678	0.4949	0.4818
	L2X-MT	0.6	0.1	0.1	0.8711	**0.6564**	**0.6465**	**0.4928**	**0.5029**	**0.4979**
M-MNIST	LR				0.9840	0.8956	0.6992	0.1888	0.4608	0.3525
	MT		0.2	0.2	**0.9851**	0.8647	0.6735	0.2379	0.4904	0.3896
	L2X	0.4			0.9593	0.5435	0.5764	0.6186	0.5877	0.6038
	L2X-MT	0.4	0.4	0.1	0.9596	**0.5291**	**0.5420**	**0.6318**	**0.6201**	**0.6260**
HAM10K	LR				0.6995	0.5757	0.6256	0.5282	0.4877	0.5093
	MT		0.3	0.2	**0.7072**	0.5749	0.6249	0.5310	0.4902	0.5119
	L2X	0.2			0.6861	0.5384	0.6045	0.5519	0.5018	0.5290
	L2X-MT	0.6	0.4	0.4	0.6861	**0.5376**	**0.6017**	**0.5525**	**0.5040**	**0.5303**

rejected and we can say that our proposed method yields high effectiveness as privacy agnostic approach LR on the primary task. One can also observe that although there is no difference between the respective performances of L2X-MT and L2x (p-value is greater than 0.05) methods with respect to the primary task on all the three datasets, but there are differences in performance of L2X-MT with LR-TFIDF and MT with respect to the primary task (p-value is less than 0.05) on TrustPilot and M-MNIST datasets respectively. On the other-hand, on each of the datasets, the p-values with respect to both the 1^{st} and 2^{nd} adversarial tasks of L2X-MT method against all the baseline methods under consideration is less than 0.05, i.e., in each of these cases, we can reject the respective null hypothesis that the performance of L2X-MT method does not differ with the baseline methods.

7.5.2 Parameter Sensitivity Analysis

We also investigate the effects of varying τ (subspace selection), and the relative importance of the adversarial task (γ_m) parameters (Equations 7.2 and 7.4) on the overall effectiveness of privacy-preservation learning of the corresponding primary tasks. Figure 7.3 shows that L2X-MT outperforms the baselines consistently for a range of

Table 7.2: Results of McNemar's test for comparing the baseline methods with our proposed method (L2X-MT).

Dataset	Comparison (L2X-MT vs.)	Primary task		1^{st} Adversarial task		2^{nd} Adversarial task	
		χ^2	p-value	χ^2	p-value	χ^2	p-value
TrustPilot	LR	0.6282	0.4280	7.9555	0.0048	4.9055	0.0268
	LR-TFIDF	5.5379	0.0186	6.6418	0.0100	4.4995	0.0339
	MT	0.3822	0.5364	4.8564	0.0275	5.5317	0.0187
	L2X	0.4193	0.5172	6.0513	0.0139	5.3376	0.0209
M-MNIST	LR	1.1830	0.2768	4.2851	0.0384	3.9934	0.0457
	MT	3.9754	0.0462	4.4692	0.0345	5.2274	0.0222
	L2X	2.7513	0.0972	6.1010	0.0135	4.4171	0.0356
HAM10K	LR	3.4994	0.0614	8.7898	0.0030	6.0839	0.0136
	MT	2.8698	0.0903	5.9377	0.0148	4.2473	0.0393
	L2X	2.8832	0.0895	7.6080	0.0058	8.0745	0.0045

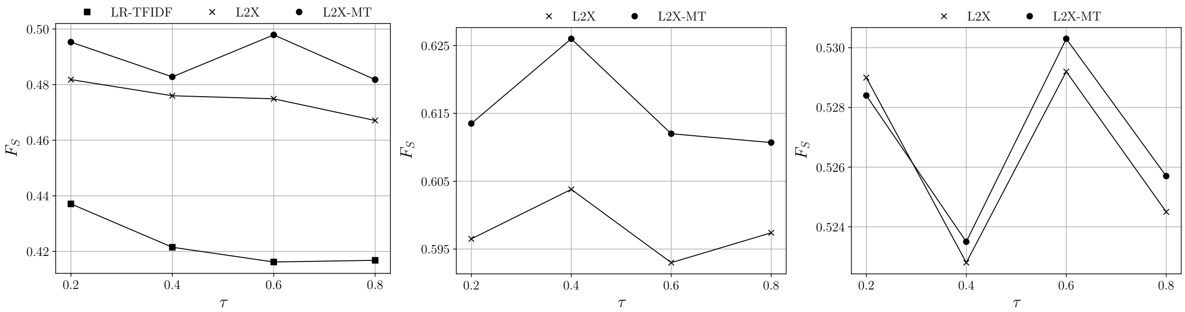
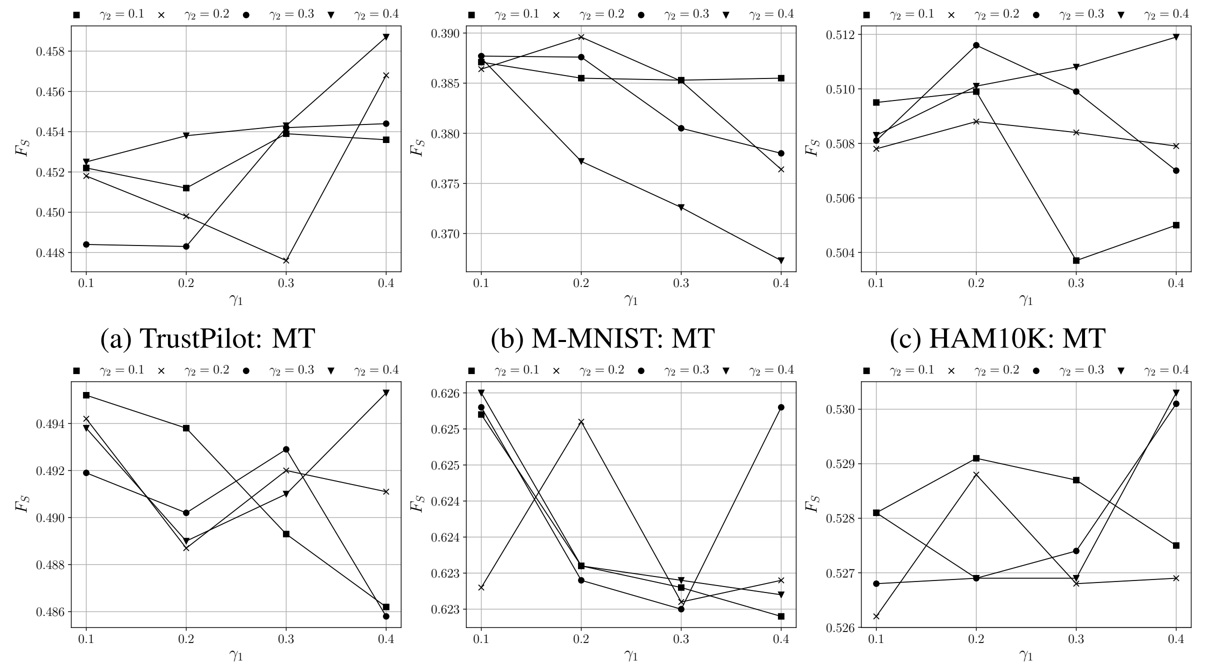


Figure 7.3: Sensitivity ((in terms of F_s)) of the privacy-aware learning approaches with respect to relative subspace dimensionality τ; Different databases used for the plots from left to right are TrustPilot M-MNIST and HAM10K respectively.

different subspace dimensions. Figure 7.4 shows the relative comparisons between the two multi-tasking approaches - MT and L2X-MT. It can be seen that for a range of different relative importance of the two adversarial tasks (e.g. age/gender detection for Trustpilot and HAM10K, and slant/broken detection for M-MNIST), leveraging information from informative subspaces helps improve the overall balance between primary task effectiveness and prevention of information leakage.

In summary, our experimental results presented in the Table 7.1 and Table 7.2 revealed the following two key observations.

1. Learning on data encoded by our method yields comparable results with that obtained on data in its original form, i.e. *our proposed encoding does not lead to a remarkable decrease in the effectiveness of a classification model.*



(a) TrustPilot: MT (b) M-MNIST: MT (c) HAM10K: MT

(d) TrustPilot: L2X-MT (e) M-MNIST: L2X-MT (f) HAM10K: L2X-MT

Figure 7.4: Sensitivity of MT, L2X-MT with variations in relative importance of two adversarial tasks.

2. Data encoded by our method *considerably reduces the effectiveness of an adversarial classification model* which seeks to predict sensitive attributes from the data. It is also shown that the use of the *informative subspace helps to improve the defence mechanism*, i.e., it further reduces the effectiveness of the adversarial classification model.

7.6 Conclusions

We proposed a generic method of privacy-preserving supervised learning, which is potentially beneficial for distributing an encoding of the input data over a cloud environment with the end-goal of eventually learning a predictive model (primary task) on the data. Our generic methodology combines the advantages of two main hypotheses - that of

a) using a *multi-task objective* that in addition to learning the primary task also learns the complementary (inverse) characteristics of an adversarial model as a defence mechanism against information stealing attacks; and

b) using a *residual subspace* of the data to further improve the defence mechanism.

Our experiments on image and text data demonstrated that our proposed method, which jointly learns a multi-objective encoding over informative subspaces (with respect to the primary task), outperforms a separate application of each.

Chapter 8

Privacy Aware Approximate Nearest Neighbour Search [*]

In the preceding three chapters, research questions RQ-1, RQ-2 and RQ-3 respectively focused on unsupervised, weakly supervised and supervised learning paradigms under the constraint of privacy preservation were discussed in details. In this chapter, we shall address the last research question RQ-4, which is

"*What is the feasibility of applying approximate nearest neighbour (ANN) based indexing and retrieval approaches under privacy preservation constraints to obtain a list of top-k suspected users, who might be infected by an infectious disease, in real time during pandemic?*"

During the Covid-19 pandemic scenario of recent past, it had been a pressing need for the administration to trace the susceptible people at the earliest who might had been infected by the virus due to their close proximity with people who were already tested positive towards the infection. This early contact tracing is important to control the otherwise unabated rate of increase in the number of infections within a locality. In this work, we investigate how effectively and efficiently can such a list of susceptible people be found given a list of infected persons and their locations. By using the locations of the given list of infected persons as queries, we investigate the feasibility of applying *approximate nearest neighbour* (ANN) based indexing and retrieval approaches to obtain a list of top-k suspected users in real-time. Since lever-

[*] Some material from [17] has been reused in this chapter.

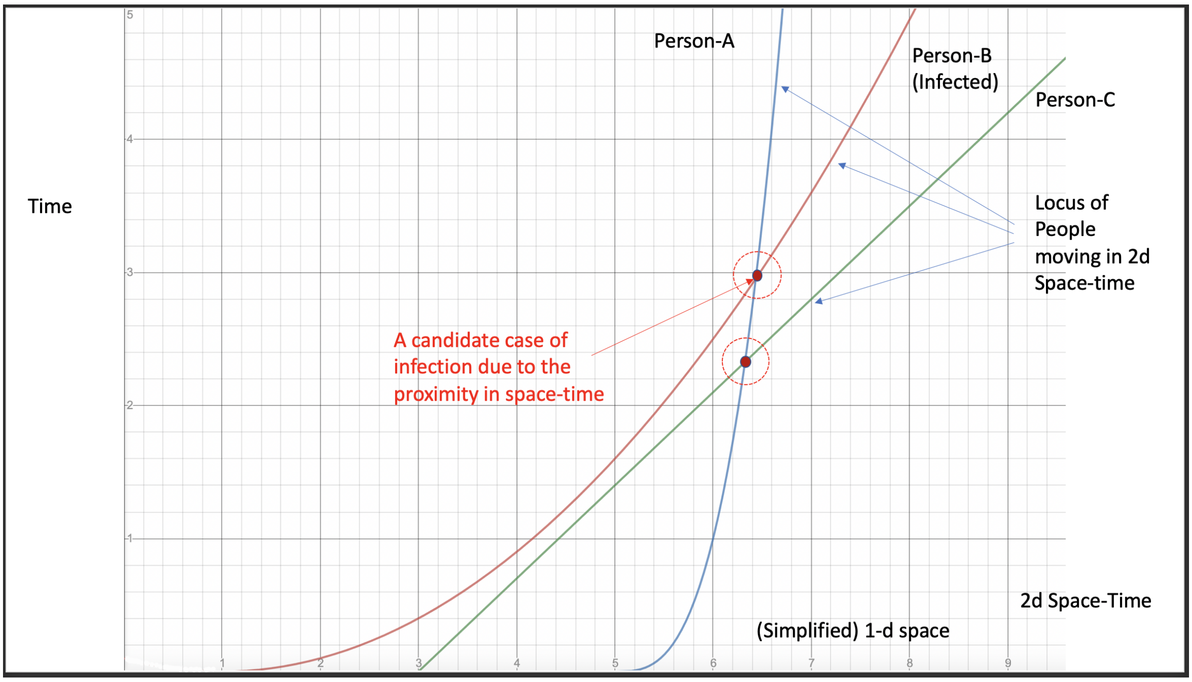

Time
Person-A
Person-B (Infected)
Person-C
Locus of People moving in 2d Space-time
A candidate case of infection due to the proximity in space-time
2d Space-Time
(Simplified) 1-d space
1
2
3
4
5
1
2
3
4
5
6
7
8
9

Figure 8.1: A simple visualization of a 2D space-time world.

aging information from true user location data can lead to privacy concerns, we also investigate the effectiveness of the ANN methods on privacy-aware encoding of the input data. Experiments conducted on real and synthetic datasets demonstrate that the top-k susceptible users retrieved with existing ANN approaches (KD-tree and HNSW) yield satisfactory recall values and achieves up to $21000\times$ speed-gain compared to exhaustive search, thus indicating that ANN approaches can potentially be applied, in practice, to facilitate real-time contact tracing even under the presence of imposed privacy constraints.

8.1 Introduction

The currently ongoing Covid-19 pandemic has spread at a rapidly accelerating rate since its inception. Standard epidemiological analysis models, e.g., the SIR model [147], have stressed on the importance of finding the susceptible cases to flatten the growth rate of the spread of infection as early as possible. In this modern era of ubiquitous digital connectivity through mobile devices, a possible source of information for contact tracing is the log of location traces in the form of GPS coordinates.

Since procuring such data for the purpose of contact tracing and using it in a restricted way (possibly by government organizations) is difficult and time-consuming due to the very sensitive nature of the data, a strong case needs to be made that how could such data be useful for controlling the spread of a pandemic. The aim of this article is to demonstrate a *proof-of-the-concept* that with the availability of massive amounts of trajectory data, it is *feasible* to develop a *scalable system* that is both effective (in terms of identifying people susceptible to an infectious disease) and efficient (in terms of the time taken to identify the susceptible cases). We believe that this proof-of-the-concept will encourage sharing (with restricted use) of such sensitive data in order to help mitigate epidemic situations.

In this work, we formalize contact tracing as a search problem in an Euclidean vector space. More concretely, each *state* of all persons is represented as a point in a 4 dimensional vector space consists of 3 dimensions for space (3 Cartesian coordinates corresponding to the spherical coordinates for latitude and longitude on the Earth's surface) and 1 for time. A given set of persons (those diagnosed as positive with the disease) then constitutes the query points in this vector space. People who were *close* to these infected persons, in terms of both space and time (i.e. they were in approximately the same place at nearly the same time), also carry the risk of being infected with the disease. The objective is to obtain a list of such susceptible people in real-time. Figure 8.1 schematically depicts the idea.

The number of points represented in this vector space can rapidly grow in situations where either the geographic area represented is too large or too dense to start with, or the location traces need to be represented over a large duration of time (e.g. over several months). An exhaustive search for finding susceptible infection cases in this space is likely not to be feasible in terms of computation time. However, this formulation makes provision to investigate the use of approximate nearest neighbor (ANN) approaches, such as KD-trees [133], and evaluate the effectiveness of such approximate approaches mainly in terms of relative recall with respect to the exhaustive search (i.e. how many such truly susceptible cases can the approximate algorithm find out). Ideally speaking, we could consider an ANN algorithm to be working well in this situation of contact tracing if it achieves a fair trade-off between the computation time and the recall relative to the exhaustive search (minimizing the former and maximizing the latter).

Our Contributions.

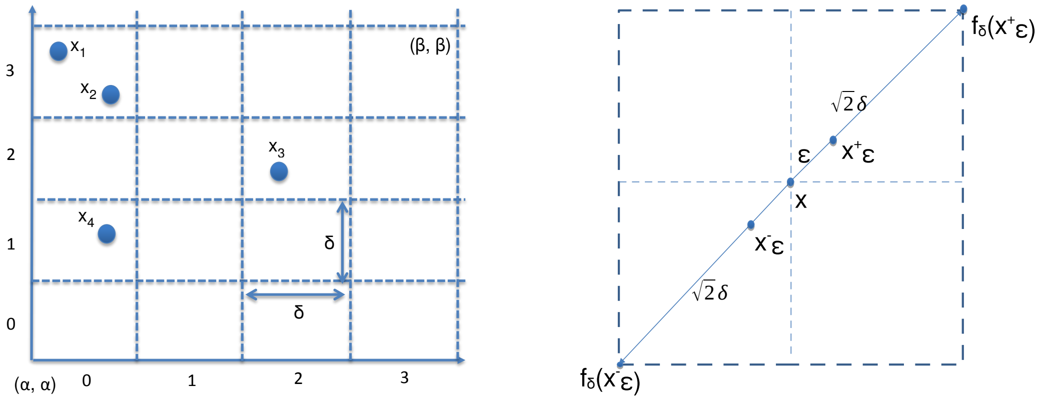
The novelty of our work lies in investigating, under a laboratory based reproducible environment, the feasibility of ANN algorithms for contact tracing during epidemics. In particular, we conduct extensive experiments on a relatively large database (24M) of real GPS locations, and an even larger collection (150M) of synthetic data comprising random walks of simulated agents. The workflow of our experiments involves indexing a large collection of trajectory records, followed by simulating a number of records from this index as infected (representing the real-life situation of new cases of reported infection). Given the location trace of each infected person, we then find out a candidate list of persons and evaluate the retrieval effectiveness. Additionally, since sharing true location data of real users across different organizations can potentially cause privacy concerns, we also investigate the feasibility of encoding the true locations with a distance-preserving linear transformation, e.g. [68]. While such encoding has been shown to preserve privacy of data [19], we investigate the effectiveness of the ANN retrieval algorithms on such an encoding.

The findings of our experiments indicate that ANN based approaches do yield satisfactory recall even on encoded data. In terms of run-time, the ANN based approach achieves up to $21000\times$ speed-up compare to exhaustive search. We emphasize that the scope of this work is not to explore a novel ANN method but rather to study the feasibility of applying ANN methods for contact tracing in an epidemic situation.

8.2 ANN Workflow

8.2.1 Representation of Location Data

The geo-locations of users (which in real life can be obtained from GPS locations of smart phones) are, in our work, represented by '3'-dimensional points (*2 space dimensions* corresponding to the location on the Earth's surface latitude, longitude formatted as $(lat, long)$ and *a time dimension* measured in system epochs). The path



(a) (b)

Figure 8.2: (a) Decomposing $\mathbb{R}^p$ into a set of l_∞ balls, (b) Maximum error in distance approximation.

traced in this 3 dimensional space-time corresponds to the activity phase of a single user. Figure 8.1 shows a schematic visualization of a 2D space-time world. Each person is shown as a path (curve) in this space-time, i.e. locus of changing positions (x coordinate) with respect to time. Figure 8.1 shows two intersections of these curves. One of these is an intersection of a healthy person with an infected one (leaving the healthy person at a high risk of infection). The objective of the ANN based search is to *automatically find all such possible intersections* given a large collection of each individual's location traces (curves in the space-time) and a given list of infected people (query curves like red one shown in the figure).

For indexing, we transform the 2-dimensional spherical coordinates of the geospatial data formatted as $(lat, long)$ into the 3-dimensional Cartesian co-ordinate system (x, y, z) using the following standard formula.

$$\begin{aligned} x &= R \cdot cos(\frac{\pi}{180} \times lat) \cdot cos(\frac{\pi}{180} \times long) \\ y &= R \cdot cos(\frac{\pi}{180} \times lat) \cdot sin(\frac{\pi}{180} \times long) \\ z &= R \cdot sin(\frac{\pi}{180} \times lat) \end{aligned} \tag{8.1}$$

In Equation 8.1, R is the radius of Earth (with an approximate value of $6,371$ km). The time dimension is appended to the 3 spatial dimensions to yield a 4-dimensional space-time data (x, y, z, t).

8.2.2 Encoding the Locations for Privacy Preservation

For contact tracing purposes, the location traces of each user over a range of time (4-dimensional space-time data) needs to be assimilated in a database. This is likely to raise privacy concerns as mandated by various privacy regulation practices, e.g. the GDPR [3]. A possible approach to prevent any possible misuse of the user's true location data is to apply a linear transformation of the data using random projections [6]. For privacy preservation, as a part of the general workflow, we first apply a distance preserving transformation function ϕ comprised of projections along random basis vectors followed by application of a quantization function, f_δ, on the projected values.

Distance-preserving transformation

Let ϕ denote the transformation function which maps points from $\mathbb{R}^d$ to its corresponding images in $\mathbb{R}^p$, i.e., $\phi : \mathbf{w} \in \mathbb{R}^d \mapsto \mathbf{x} \in \mathbb{R}^p$. The most common function for such transformation is the locality sensitive hash function (LSH) [6], which involves randomly choosing a set of p basis vectors $\mathfrak{B}$, where p is a parameter. Each point is then transformed by computing projections of the point along these p basis vectors yielding the p components of the transformed point in $\mathbb{R}^p$. More concretely, the i^{th} component of the transformed vector in $\mathbb{R}^p$ is given by

$$x_i = \mathbf{w} \cdot \mathbf{b}_i, \tag{8.2}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a (raw) data vector (e.g. the true user trajectories), and $\mathbf{b}_i \in \mathfrak{B}$ is the i^{th} basis vector (detail description about the LSH transformation is given in the Section 3.2).

A random basis ensures that computing the inverse function is non-tractable [6]. However, as per the Johnson-Lindenstrauss (JL) lemma [161], it is known that this random projection based transformation of Equation 8.2 is in fact distance preserving [167]. The robustness of this distance preserving transformation is further improved in [68] by applying orthogonalization on the randomly chosen basis vectors using Gram-Schmidt method. In this work, we specifically use the orthogonal basis vector based approach of [68] as a definition of the transformation function ϕ.

Quantizing the projections

The purpose of quantization of the projected values is two fold. First, quantizing the projected values adds a further layer of obfuscation on the projected values. Second, it helps to reduce the storage space (4 or 8 bytes of floating point vs. a single byte which allows for up to 256 possible quantized values) and hence allows faster loading of parts of the index into the main memory thereby speeding up the retrieval process.

The key idea in quantization is to transform the real-valued Cartesian space, $\mathbb{R}^p$, into a set of non-overlapping axis-parallel grids. More formally, each grid represents an l_∞ ball of some positive radius $\delta \in \mathbb{R}$, taking the shape of a hyper-cube of length δ. This transformation is visualized for the particular case of 2 dimensions for 4 points, $\mathbf{x_1}, \ldots, \mathbf{x_4}$ in Figure 8.2a, where each l_∞ ball manifests itself as a square cell.

If $X = \bigcup_{i=1}^{N} \{\mathbf{x_i}\}$ denotes a set of N points in $\mathbb{R}^p$, to place the grid over X, we first calculate the length of each grid, denoted by δ. The value of δ is a function of a) the number of equi-spaced intervals M, in which we would want to split each basis vector (axis), and b) the minimum and the maximum coordinates along the axes, denoted by α and β respectively. Thus,

$$\delta = \frac{\beta - \alpha}{M}, \quad \alpha = \min_{i=1}^{N} \min_{j=1}^{p} \mathbf{x_i}_j, \quad \beta = \max_{i=1}^{N} \max_{j=1}^{p} \mathbf{x_i}_j \tag{8.3}$$

The l_∞ balls are hence centred at points $\mathbf{c} \in \mathbb{R}^p$, where

$$\mathbf{c} = \{\alpha + (r + \frac{1}{2})\delta\}^p, \quad r = 0, \ldots, M-1 \tag{8.4}$$

We then define a transformation function, $f_\delta(\mathbf{x})$, which represents a point $\mathbf{x}$ by the coordinates of its discrete grid locations along each dimension. More formally,

$$f_\delta(x_i) = \left\lceil \frac{x_i - \alpha}{\delta} \right\rceil, \quad \forall i = 1, \ldots, p. \tag{8.5}$$

The distance between two quantized points is given by

$$\mathcal{D}_\delta(f_\delta(\mathbf{x}), f_\delta(\mathbf{y})) = \Big(\sum_{i=1}^{p} \big(\left\lceil \frac{x_i - \alpha}{\delta} \right\rceil - \left\lceil \frac{y_i - \alpha}{\delta} \right\rceil\big)^2\Big)^{\frac{1}{2}} \tag{8.6}$$

Figure 8.2b demonstrates the approximation effect of the quantization in two dimensions. Maximum quantization error occurs when two points, $\mathbf{x}_{\epsilon}^{+}$ and $\mathbf{x}_{\epsilon}^{-}$, in the ϵ-neighbourhood of $\mathbf{x}$ are transformed to two different points $f_{\delta}(\mathbf{x}_{\epsilon}^{+})$ and $f_{\delta}(\mathbf{x}_{\epsilon}^{-})$ respectively. The separation distance between these two transformed points in two dimensions is $\sqrt{2}\delta$, whereas for the general case of p dimensions, this distance is $\sqrt{p}\delta$. Hence, the maximum factor by which distances are magnified, in the general case of p dimensions, is given by

$$\frac{\mathcal{D}(f_{\delta}(\mathbf{x}_{\epsilon}^{+}), f_{\delta}(\mathbf{x}_{\epsilon}^{-}))}{\mathcal{D}(\mathbf{x}_{\epsilon}^{+}, \mathbf{x}_{\epsilon}^{-})} = \frac{2\sqrt{p}\delta}{2\epsilon} = \frac{\sqrt{p}\delta}{\epsilon} \tag{8.7}$$

As expected, this distortion can be reduced with small values of δ, which is a parameter of the quantization process. In other words, the closer a point is to the corner point between two grids, i.e. lower the value of ϵ, the higher is the quantization error.

8.2.3 Retrieval of Susceptible Cases

The quantized ϕ trasformed data, $\{f_{\delta}(\phi(\mathbf{w}) : \mathbf{w} \in \mathbb{R}^{d}\}$, is then either stored in the memory (for the KD-tree approach) or saved into an index (for the HNSW approach). The next step is to retrieve the susceptible cases. We simulate the case that a fraction of the population (whose data exists in the index already) has been infected.

The 'retrieval' procedure formulates and executes a query for each of these infected people and reports a list of K (K is a parameter) most susceptible persons that came in close contact (in terms of space and time) with an infected person.

We perform the retrieval procedure using HNSW or KD-tree searching algorithm (discussed in the Section 3.7). The HNSW and KD-tree algorithm conducted on distance-preserving ϕ-transformed (abbreviate as 'DPT') data are named as 'DPT-HNSW' and 'DPT-KD-tree' while the algorithms executed on privacy-preserved (abbreviate as 'PP') encoding of data, $\{f_{\delta}(\phi(\mathbf{w})\}$, are named as 'PP-HNSW' and 'PP-KD-tree'.

8.3 Experimental Setup

8.3.1 Dataset

To study the effectiveness of our Approximate Nearest Neighbour Search system we perform a number of experiments with both real and synthetic datasets. As a real dataset, we use the FourSquare[2] global check-in dataset. To conduct experiments on a yet larger collection, we simulate synthetic trajectories, with a different number of users (simulated agents) and number of time steps (range of time). The detail description about these real and synthetic datasets are presented in the Section 4.3.

8.3.2 Parameters and Evaluation Metrics

The two main parameters for privacy preserving data encoding are p, the dimension of the set of basis vectors $\mathfrak{B}$ for the transformation ϕ and M, the number of quantization grids which is inversely proportional to the quantization interval δ (c.f. Equation 8.3 and 8.4). For our experiments, we set the values of p as 2, 4, 8, and 16. M is chosen independently for each dataset depending on the population density ρ. For Traject-10K and Traject-100K datasets we set M to 16, 32, 64 and 128, whereas for Traject-1M data we set it to 128, 256, 512 and 1024. Likewise for the FourSquare dataset (CheckIn-24M), we set its value to $1K$, $10K$, $100K$ and $1M$.

For synthetic trajectory datasets, we conduct ANN retrieval for each space-time coordinate of an infected (query) user. This means that a final list of susceptible candidates is obtained by aggregating (set union) of these individual lists. The number of retrieved candidates, say r (#retrieve/timestep), at each distinct time coordinate value is varied from 10 to 100 in steps of 10.

Since the task of finding susceptible candidates is a recall-oriented task (false negatives are less desirable), we evaluate the effectiveness of susceptible retrieval with recall, which measures the proportion of the true nearest neighbors (true susceptible candidates) that are eventually retrieved.

[2] https://drive.google.com/file/d/0BwrgZ-IdrTotZ0U0ZER2ejI3VVk/view

Table 8.1: Summary of the results of experimentation of Approximate Nearest Neighbour retrievals on synthetic and real datasets. Here, #retrievals per time step (r) for each infected user is set to 100, p is the dimension of ϕ-encoding space $\mathbb{R}^p$, M is the number of quantization grids for the transformation function $f_\delta(\mathbf{x})$ and the retrieval time t is measured in milliseconds (ms).

	Exhaustive		HNSW		DPT-HNSW			PP-HNSW				KD-tree		DPT-KD-tree			PP-KD-tree			
Dataset	t (ms)	Recall	t (ms)	Recall	p	t (ms)	Recall	p	M	t (ms)	Recall	t (ms)	Recall	p	t (ms)	Recall	p	M	t (ms)	Recall
Traject-10K	5294.95	1.00	1.91	0.9796	16	1.82	0.9706	16	128	1.10	0.9275	269.74	0.9823	16	750.46	0.9798	16	128	716.44	0.9821
Traject-100K	24813.08	1.00	3.83	0.9601	16	3.25	0.9410	16	128	2.19	0.8435	270.07	0.9653	16	656.78	0.9637	16	128	1068.63	0.9545
Traject-1M	524813.08	1.00	28.25	0.4610	16	19.19	0.4418	16	1024	24.40	0.4384	1242.05	0.7952	16	1874.97	0.7974	16	1024	1623.07	0.7830
CheckIn-24M	10408.12	1.00	0.01	0.9983	16	0.02	0.7660	16	1M	0.02	0.7428	1.91	1.0000	16	6.34	0.7888	16	1M	6.60	0.7893

Table 8.2: Results of McNemar's hypothesis testing for comparing the Approximate Nearest Neighbour based retrieval approaches using both synthetic and real datasets.

	Traject-10K		Traject-100K		Traject-1M		CheckIn-24M	
Comparison	χ^2	p-value	χ^2	p-value	χ^2	p-value	χ^2	p-value
HNSW vs. KD-tree	0.4058	0.5241	3.2948	0.0695	5.5577	0.0184	1.7037	0.1918
HNSW vs. DPT-HNSW	0.6630	0.4155	0.0064	0.9361	0.3327	0.5641	5.6759	0.0172
HNSW vs. PP-HNSW	6.0508	0.0139	4.3689	0.0366	0.3826	0.5362	9.6437	0.0019
DPT-HNSW vs. PP-HNSW	4.7855	0.0287	7.0335	0.0080	0.1463	0.7021	3.0915	0.0787
KD-tree vs. DPT-KD-tree	1.2816	0.2576	0.1592	0.6899	1.6409	0.2002	4.6922	0.0303
KD-tree vs. PP-KD-tree	0.8396	0.3595	2.0682	0.1504	0.0045	0.9464	6.5312	0.0106
DPT-KD-tree vs. PP-KD-tree	0.0412	0.8391	0.6359	0.4252	1.2404	0.2654	0.3731	0.5413
DPT-HNSW vs. DPT-KD-tree	2.0825	0.1490	0.4789	0.4889	6.1421	0.0132	6.2528	0.0124
PP-HNSW vs. PP-KD-tree	5.8931	0.0152	4.3830	0.0363	5.4294	0.0198	4.3830	0.0363

8.4 Results

8.4.1 Summary of the Results

Table 8.1 present the results of the different ANN search workflows. The key observations from the table are following.

First, we observe that both the approaches HNSW and KD-tree yield satisfactory recall values which demonstrates the feasibility of applying an ANN-based workflow in pandemic situations to achieve a trade-off between recall and computation time. The time, t (ms), reported in milli-seconds refers to the time taken to retrieve a list for a single query.

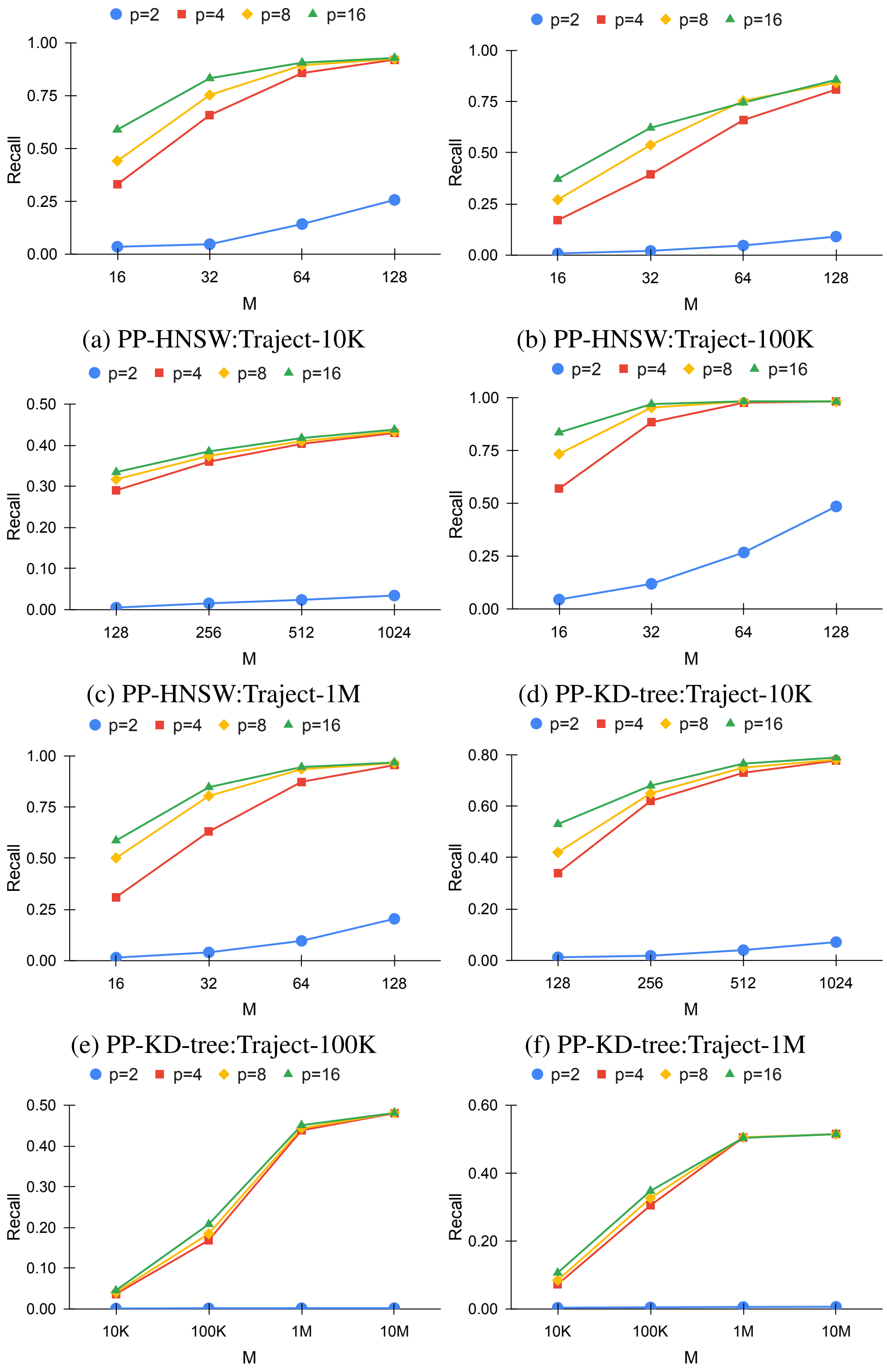
The retrieval times of both KD-tree and HNSW are substantially lower than an exhaustive search through the database (for the Traject-1M dataset, the exhaustive search takes $21000\times$ more time on an average). Although KD-tree, DPT-KD-tree and PP-KD-tree yield better recall values than HNSW, DPT-HNSW and PP-HNSW respectively, the retrieval times required by KD-tree based ANN approaches (e.g. KD-tree, DPT-KD-tree and PP-KD-tree) are higher than the same of HNSW based ANN approaches (e.g. HNSW, DPT-HNSW and PP-HNSW).

As in the preceding Chapter, here also we have conducted the McNemar's test (χ^2 test) (refer to Section 4.4.2 for details) to compare the retrieval accuracies of different ANN-based approaches. The null hypothesis has been formulated as the performance of two retrieval methods (which are under consideration) are same. The significance threshold has been set at 0.05, i.e., if the observed p-value is found to be less than the chosen significance level, the null hypothesis gets rejected. Table 8.2 presents the caparative results based on McNemar's test on different synthetic and real datasets. Here, it may be observed that the p-values of the McNemar's test for HNSW vs. KD-tree are greater than 0.05 on all the datasets under consideration barring only the Traject-1M dataset. Such an observation is somewhat true for all other comparisons included in the same Table. Thus, it may be concluded that there is no significant difference in the performance of relevant privacy preserving computational approaches.

8.4.2 Parameter Sensitivity Analysis

Figure 8.3 presents the sensitivity of two parameters for privacy preserving data encoding namely p, dimension of set of basis vectors $\mathfrak{B}$ and M, number of quantization grids on PP-HNSW and PP-KD-tree. From the figure, we observe that increasing the number of quantization grids increases recall values. However, we also note that it is not required to increase the value of M in some arbitrary way because the results tend to saturate out with the use of $M = 128$ for Traject-10K and Traject-100K datasets and the value $M = 1024$ for Traject-1M dataset. Since the density of Check-in dataset is higher than those of the synthetic ones (see Table 4.3), the number of grids required to achieve satisfactory recall values is also higher for this dataset and Figure 8.3 shows that about 1M grids required in both PP-HNSW and PP-KD-tree to obtain satisfactory recall values.

In Figure 8.4, we observe that by increasing the value of r, we can obtain better recall value. However, this leads to an increase in the retrieval time and we have also found that $r = 100$ gives near optimal results in all datasets within satisfactory re-



(a) PP-HNSW:Traject-10K

(b) PP-HNSW:Traject-100K

(c) PP-HNSW:Traject-1M

(d) PP-KD-tree:Traject-10K

(e) PP-KD-tree:Traject-100K

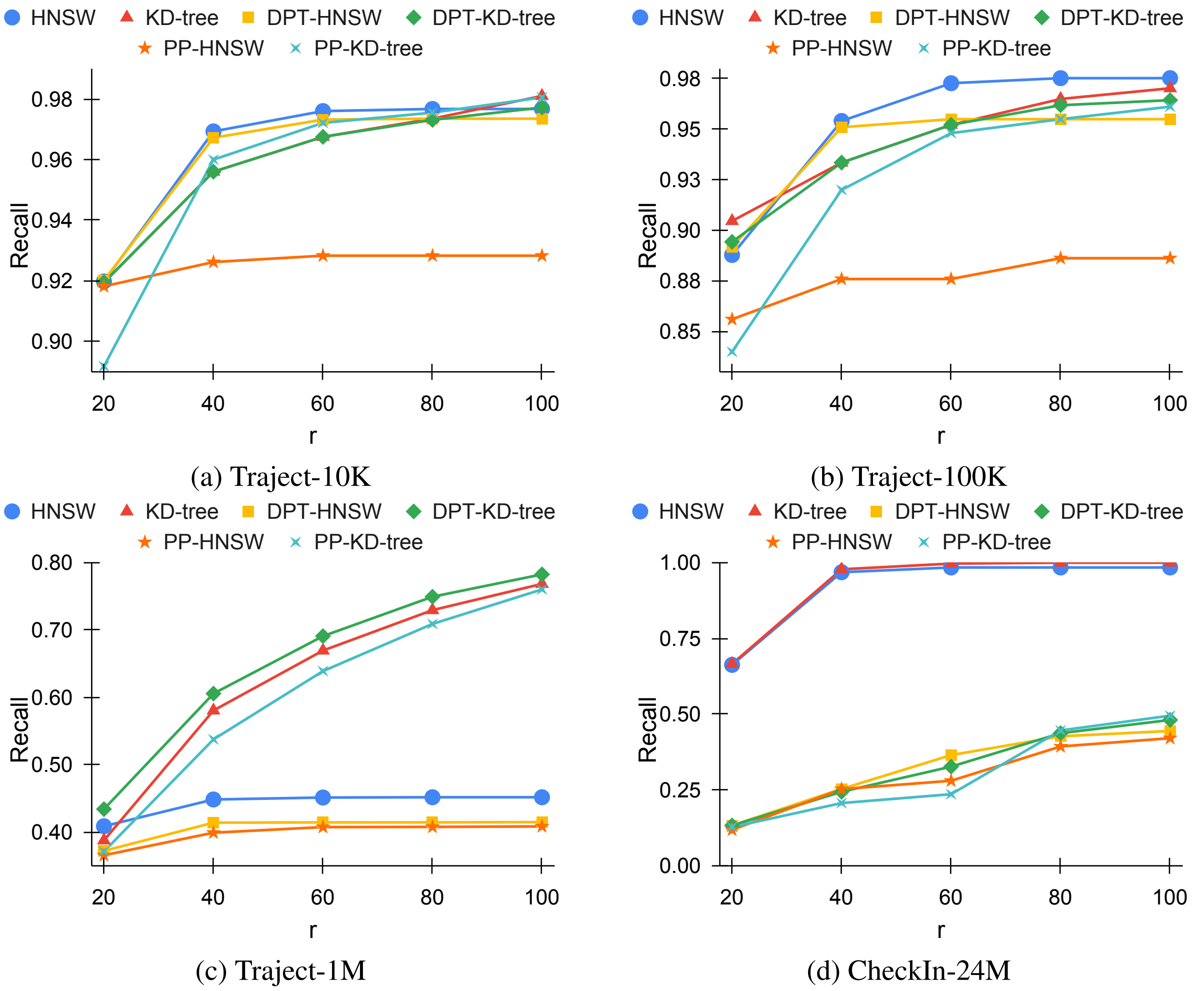
(f) PP-KD-tree:Traject-1M

(g) PP-HNSW:CheckIn-24M

(h) PP-KD-tree:CheckIn-24M

Figure 8.3: Sensitivity of ANN retrieval effectiveness with variations in the projection dimension (p) and the number of quantization grids (M).

trieval time.



(a) Traject-10K

(b) Traject-100K

(c) Traject-1M

(d) CheckIn-24M

Figure 8.4: Sensitivity of ANN retrieval effectiveness with respect to the number of retrieved users (r) at each timestep.

8.5 Conclusions

In this chapter, we investigated the feasibility of applying standard approximate nearest neighbor (ANN) search approaches for the task of contact tracing in pandemic situations. More concretely, given an indexed collection of space-time coordinates of individuals and a list of infected persons, our task is to retrieve a list of candidate persons that might be susceptible to the infection since they came in close proximity (approximately same place and time) with the people already infected. Since location data for contact tracing could lead to privacy issues, we also propose an encoding and quantization based obfuscation of the data.

We conduct a set of laboratory-based experiments on data with known groundtruths. We found that the recall values that could be achieved with ANN-based approaches are satisfactory. Although the recall levels do decrease with an increase in the number of data points, our experiments show that for large datasets ANN based retrieval can achieve speed-gains of up to $21000\times$, thus achieving a relative trade-off between run-time and accuracy. These savings in run-time could be pivotal for early identification of susceptible cases and carry out necessary measures (e.g. quarantine the susceptible persons) for the health-care safety of a community. The proposed workflow also ensures that it is not required to share true user locations for contact tracing purposes. Instead, such a methodology for contact tracing in pandemic situations works fairly well with distance-preserving transformation of the data.

Chapter 9

Conclusions and Future Work

In this thesis, we have presented reports of our study towards preservation of data privacy in performing two popular machine learning (ML) tasks such as clustering and classification over cloud platforms. Though, preservation of privacy is not any significant issue to be studied separately for performing these ML tasks at home, office or similar environments, the situation is different when the computing platform is based on the cloud. In cloud environment, data is required to be sent in encoded form to ensure prevention of any sensitive information leakage. It may so happen that although some ML algorithms produce better results on certain source data but the same fail to produce similar results when the data gets encoded before feeding them as input to the ML tasks. In such cases, users need to search for privacy preserving ML algorithms that are capable of providing acceptable results on the encoded data compared to the similar results obtained on the corresponding source data. Thus, a trade-off between the performance of the ML model and the chances of sensitive information leakage of the data is required to be adopted. In this last chapter of the present thesis, we conclude our studies by providing an overall analysis of the findings related to the four research questions formulated by us in the beginning. Finally, we discuss about an outline of possible future work.

9.1 Research Questions Revisited

In this section, we revisit the four research questions introduced by us in Chapter 1 of this thesis and present a summary of the solution proposed by us for each of them in the previous chapters.

9.1.1 Privacy Aware Unsupervised Learning

The motivation behind developing the privacy preserving unsupervised learning algorithm, in particular privacy preserving approximate K-means clustering algorithm was some important characteristics of the Hamming space, such as:

- We need much smaller memory for storing the Hamming space transformed vector compare to the real valued representation of that vector.
- The inner product based similarity measure of two binary vectors in the Hamming space is very less expensive than the similarity measure using floating point operation in Euclidean space.
- The Hamming space representation preserves the privacy of the data in the sense that it is computationally very difficult to obtain the real valued inverse representation of the transformed binary vectors.

The first research question, RQ-1, introduced in Chapter 1 is following:

> **RQ-1:** *How unsupervised learning algorithm can be re-designed under the constraint of privacy preservation to improve the learning effectiveness?*

The objective of our first research question RQ-1 was to propose a clustering algorithm on Hamming space which approximates the K-means clustering on the Euclidean space and it preserves the data privacy. In-particular we have used the Hamming space transformation for privacy preservation of the data and proposed a novel approach for centroid re-computation of K-means algorithm on the Hamming space. In the proposed algorithm we have used the projection statistics along the basis vectors (used in LSH) collected at the time of Hamming space transformation.

The first research question RQ-1 is thus addressed by proposing a novel privacy preserving approximate K-means clustering algorithm. From the proposed solution of RQ-1 we can conclude that the unsupervised learning can be performed under privacy preservation constraint by executing it on the Hamming space and the standard K-means clustering algorithm can be re-design to better perform under privacy preservation constraint.

9.1.2 Privacy Aware Semi-Supervised Learning

Successful exploration of the first research question RQ-1 on unsupervised learning motivated us to study semi-supervised learning algorithm under privacy preservation constraints to achieve a better result that the unsupervised learning. Thus, the second research question, RQ-2, on privacy preserving semi-supervised learning, introduced in Chapter 1, is given by:

> **RQ-2:** *How the effectiveness of privacy preserving clustering on discrete metric space can be improved with weak supervision on the encoding transformation?*

We call the proposed algorithm of the solution of RQ-2 as a semi-supervised clustering algorithm in the sense that, we are using a fraction of original data to learn a dense encoding of the privacy preserving binary encoding of the dataset. It is to be noted that we are not using the ground truth label of the dataset in any step of the algorithm.

The main objective of our proposed solution of the second research question RQ-2 was to generate a dense embedding of the binary encoding of input data such that the clustering algorithm on the dense representation of the data yields better results than that of on binary representation of the data. Thus, we proposed an encoding-based workflow of data clustering that preserves data privacy and it is suitable for deployment in a distributed computing environment, where most of the computation is conducted at the server side on encoded data. In our proposed solution a weakly supervised approach is used to learn a parameterized similarity function with the application of triplet networks on a small seed set of data to guide the clustering process at the server side.

9.1.3 Privacy Aware Supervised Learning

The third research question which we have explored in this thesis is about exploring the usefulness of supervised learning against information stealing attacks.

> **RQ-3:** *How supervised learning can be used to defend the malicious attempts of stealing sensitive information from data shared on cloud platforms?*

To explore the solution of this research question, in Chapter 7 we propose an informative subspace based multi-objective framework.

The primary feature of our proposed framework are as follows:

- Encode data so that the encoded representation provides resilience against information stealing attacks.
- Useful for sharing data in a software-as-a-service (SaaS) environment where the computations (e.g. parameter updates) is conducted at the server side.
- A general multi-objective based solution that is able to leverage from the most informative feature subspace to achieve an effective encoding.

Thus the primary goal of our multi-objective based approach is to generate a privacy aware encoding of the data which minimize the chances of sensitive information leakage.

We evaluated our proposed model and other baseline methods on two image datasets, namely Morpho-MNIST, Skin Cancer MNIST and one text dataset, namely Trustpilot (US English) to show the effectiveness of our proposed solution and found that our proposed solution outperforms compared to other baselines.

9.1.4 Privacy Aware Approximate Nearest Neighbour Search

After the exploration of unsupervised, semi-supervised and supervised learning under privacy preservation constraints we have explored the usefulness of applying approximate nearest neighbour under privacy preservation constraints. Thus our last research question investigated in this thesis is following:

> **RQ-4:** *What is the feasibility of applying approximate nearest neighbour (ANN) based indexing and retrieval approaches under privacy preservation constraints to obtain a list of top-k suspected users, who might be infected by an infectious disease, in real time during pandemic?*

We have proposed the solution of this research question RQ-4 in the Chapter 8. The main highlights of our proposed solution of the research question RQ-4 are following:

- Encoded the data so that the encoded representation provides resilience against information stealing attacks.
- Useful for sharing data in a software-as-a-service (SaaS) environment where the computations (e.g. parameter updates) is conducted at the server side.
- Investigated the feasibility of applying *approximate nearest neighbour* (ANN) based indexing and retrieval approaches to obtain a list of top-k suspected users with infectious disease in real-time.
- Experiments conducted on standard retrieval task using real and synthetic datasets demonstrate the efficacy of the ANN method in simulated epidemics to identify susceptible infected persons.

9.2 Future Work

While this thesis has explored various privacy aware machine learning techniques, in particular we focus on improvement of a supervised classification model in addition to the K-means clustering algorithm under privacy preservation constraints, there remain a number of scope for future work, which we believe deserves more exploration.

Chapter 5: For privacy aware unsupervised learning we have focused on the most popular clustering algorithm K-means and it is observed that some additional statistical information can improve the effectiveness of K-means clustering under privacy preservation constraints.

In future, we would like to address privacy preservation constraints for other clustering methods, e.g. DBSCAN, and also formalize the notions of differential privacy under such a setup.

Chapter 6: Similar to the unsupervised learning, for privacy aware semi-supervised learning also we have focused on the improvement of the K-means clustering algorithm in a discrete metric space and found that deep metric learning with weak supervision on a small seed set of data leading effective encoding of the data, yields better K-means clustering.

In future, we would like to explore ways of obtaining effective results on other privacy-aware machine learning tasks, e.g., reinforcement learning etc.

Chapter 7: For privacy aware supervised learning we have proposed an informative subspace based multi-objective approach to produce a privacy aware encoding of the input data where the sensitive attributes are explicitly annotated for training data and found that this encoding minimizes the privacy leakage without compromising too much on the effectiveness of the primary task.

In future studies, we would like to explore the possibility of obtaining a privacy-preservation encoding scheme of the input data in those cases where the sensitive attributes are latent rather than being manifested as explicitly annotated identifiable attributes (i.e., to address the situation when the attribute value annotations are not available in the training set). Unsupervised analysis of the input space coupled with a semi-supervised encoding approach can potentially be useful to tackle such a situation.

Chapter 8: It has dealt with the problem of locating the persons who had recently come in close contact of a person diagnosed positive of the infectious virus during a pandemic scenario using their geo-location trace. As this geo-location trace is a sensitive information of the persons involved, so we have proposed a privacy preserving strategy to perform this retrieval task. This strategy requires the data to be sent in an encoded form to the server offering computational service for the retrieval task. The proposed privacy preserving retrieval strategy has used LSH based transformation function followed by a quantization transformation to produce the encoded data for retrieval of required information based on HNSW and KD-tree. approaches

In future studies, we plan to explore differential privacy preservation strategies for the same contact tracing problem of the pandemic scenario.

9.3 Closing Remarks

It will not be an exaggeration of the fact if we claim that the study performed in this thesis has set a new direction of research in the area of privacy preserving machine learning through the exploitation of Hamming space transformation as well as the information on the importance of various components of the encoded vector representation. We hope that the results of the experimentation performed by us will encourage other researchers working in the area of privacy-preserving machine learning to continue further studies in the same direction and explore prospective new applications. A few possible applications of privacy-preserving machine learning are discussed below.

1. Privacy-preserving machine learning can be used in traffic analysis as well as route optimization without compromising the drivers' privacy.

2. In applications of NLP, text data from the users is processed to extract its insights or sentiment analysis. The goal of privacy-preserving machine learning is maintenance of the confidentiality of user identity or his/her private information in the course of processing similar data.

3. In smart grid systems, privacy-preserving machine learning can be used to analyze the power consumption patterns without revealing the sensitive information of the consumers.

4. Image and video data often contain sensitive information of the users. Privacy preserving machine learning can be used to process the images and videos without compromising individual user's identity and other sensitive information.

Bibliography

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[2] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Enriching textbooks with images. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1847–1856, 2011.

[3] Efthimios Alepis, Eugenia Politou, and Constantinos Patsakis. Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions. *Journal of Cybersecurity*, 4(1), 03 2018.

[4] Thamer Altuwaiyan, Mohammad Hadian, and Xiaohui Liang. Epic: efficient privacy-preserving contact tracing for infection detection. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[5] Giuseppe Amato, Claudio Gennaro, and Pasquale Savino. Mi-file: Using inverted files for scalable approximate similarity search. *Multimedia Tools Appl.*, 71(3):1333–1362, August 2014.

[6] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 459–468. IEEE, 2006.

[7] Z. Arkaitz and J. Heng. Harnessing web page directories for large-scale classification of tweets. In *Proceedings of the 22nd international conference on World Wide Web companion*, WWW '13 Companion, pages 225–226, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[8] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[9] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[10] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

[11] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the 14th international conference on World Wide Web*, pages 651–660, 2005.

[12] Jeffrey S Beis and David G Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1000–1006. IEEE, 1997.

[13] James Bell, David Butler, Chris Hicks, and Jon Crowcroft. Tracesecure: Towards privacy preserving contact tracing. *arXiv preprint arXiv:2004.04059*, 2020.

[14] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[15] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[16] Garrett Bernstein and Daniel R Sheldon. Differentially private bayesian linear regression. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Chandan Biswas, Debasis Ganguly, and Ujjwal Bhattacharya. Approximate nearest neighbour search on privacy-aware encoding of user locations to identify susceptible infections in simulated epidemics. In *Forum for Information Retrieval Evaluation*, pages 35–42, 2021.

[18] Chandan Biswas, Debasis Ganguly, Partha Sarathi Mukherjee, Ujjwal Bhattacharya, and Yufang Hou. Privacy-aware supervised classification: An informative subspace based multi-objective approach. *Pattern Recognition*, page 108301, 2021.

[19] Chandan Biswas, Debasis Ganguly, Dwaipayan Roy, and Ujjwal Bhattacharya. Privacy preserving approximate k-means clustering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1321–1330, 2019.

[20] Chandan Biswas, Debasis Ganguly, Dwaipayan Roy, and Ujjwal Bhattacharya. Weakly supervised deep metric learning on discrete metric spaces for privacy-preserved clustering. *Information Processing & Management*, 60(1):103109, 2023.

[21] Chandan Biswas, Partha Sarathi Mukherjee, Koyel Ghosh, Ujjwal Bhattacharya, and Swapan K Parui. A hybrid deep architecture for robust recognition of text lines of degraded printed documents. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3174–3179. IEEE, 2018.

[22] Marian Boguna, Dmitri Krioukov, and Kimberly C Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74–80, 2009.

[23] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2019.

[24] Daniel C Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-mnist: quantitative assessment and diagnostics for representation learning. *Journal of Machine Learning Research*, 20(178):1–29, 2019.

[25] Sandro Cavallari, Vincent W Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 377–386, 2017.

[26] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12):5017–5032, 2015.

[27] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[28] Kinjal Chaudhari and Ankit Thakkar. Survey on handwriting-based personality trait identification. *Expert Systems with Applications*, 124:282 – 308, 2019.

[29] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM computing surveys (CSUR)*, 33(3), 2001.

[30] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proc. of ICML '18*, pages 883–892, 2018.

[31] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1335–1344, 2016.

[32] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546. IEEE, 2005.

[33] Paolo Ciaccia and Marco Patella. PAC nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces. In *Proc. of ICDE '00*, pages 244–255, 2000.

[34] Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. Privacy-preserving neural representations of text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[35] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of HLT-NAACL '19*, pages 4171–4186. Association for Computational Linguistics, June 2019.

[37] Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, and Benjamin IP Rubinstein. Differential privacy for bayesian inference through posterior sampling. *JMLR*, 18(1):343–381, 2017.

[38] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[39] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284, 2006.

[40] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7:17–51, 2017.

[41] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[42] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390, 2009.

[43] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[44] B. Eisenberg and R. Sullivan. Why is the sum of independent normal random variables normal. In *Math. Mag.*, volume 81, pages 362–366, 2008.

[45] Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, October-November 2018.

[46] H. Chang et al. Robust path-based spectral clustering. *Pattern Recognition*, 41:191–203, 2008.

[47] J. Anil K et al. Data clustering: A user's dilemma. In *PReMI*, pages 1–10, 2005.

[48] M. Yusuke et al. PQk-means: Billion-scale clustering for product-quantized codes. In *Multimedia Conference*, pages 1725–1733, 2017.

[49] T. Mikolov et al. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS '13*, pages 3111–3119, 2013.

[50] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Proc. of NIPS '13*, pages 2121–2129, 2013.

[51] Limin Fu and Enzo Medico. Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC bioinformatics*, 8(1):1–15, 2007.

[52] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Variational information maximization for feature selection. In *Proc. of NIPS '16*, pages 487–495, 2016.

[53] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proc. of ECCV '18*, pages 269–285, 2018.

[54] J. Geetha and W. Rebecca N. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proc. of KDD '05*, pages 593–599. ACM, 2005.

[55] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proc. of ICCV '17*, pages 5736–5745, 2017.

[56] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[57] Yunchao Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 817–824, 2011.

[58] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR '15*, 2015.

[59] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proc. of CIKM '16*, pages 55–64, 2016.

[60] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[61] J. Herve, D. Matthijs, and S. Cordelia. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

[62] Kalun Ho, Janis Keuper, Franz-Josef Pfreundt, and Margret Keuper. Learning embeddings for image clustering: An empirical study of triplet loss approaches. In *Proc. of ICPR '21*, pages 87–94. IEEE, 2021.

[63] Dirk Hovy, Anders Johannsen, and Anders Søgaard. User review sites as a resource for large-scale sociolinguistic studies. In *Proc. of WWW '15*, pages 452–461, 2015.

[64] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[65] D. Irit and N. Kobbi. Revealing information while preserving privacy. In *Proc. of Symposium on Principles of Database Systems*, pages 202–210. ACM, 2003.

[66] Anil K Jain and Martin HC Law. Data clustering: A user's dilemma. In *International conference on pattern recognition and machine intelligence*, pages 1–10. Springer, 2005.

[67] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proc. of ICLR '17*. OpenReview.net, 2017.

[68] J. Ji, J. Li, S. Yan, B., and Q. Tian. Super-bit locality-sensitive hashing. In *Advances in neural information processing systems*, pages 108–116, 2012.

[69] Y. Jinfeng, W. Jun, and J. Rong. Privacy and regression model preserved learning. In *Proc. of AAAI '14*, pages 1341–1347, 2014.

[70] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[71] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S Verykios. Distance-aware encoding of numerical values for privacy-preserving record linkage. In *Proc. of ICDE '17*, pages 135–138. IEEE, 2017.

[72] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S Verykios. Federal: A framework for distance-aware privacy-preserving record linkage. *IEEE Transactions on Knowledge and Data Engineering*, 30(2):292–304, 2017.

[73] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):664–676, April 2017.

[74] H. Kashima, J. Hu, B. Ray, and M. Singh. K-means clustering of proportional data using l1 distance. In *Proc. of ICPR '08*, pages 1–4, 2008.

[75] Siddhesh Khandelwal and Amit Awekar. Faster k-means cluster estimation. In *European Conference on Information Retrieval*, pages 520–526. Springer, 2017.

[76] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of 2nd International Conference on Learning Representations, ICLR*, 2014.

[77] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. of the ACM symposium on Theory of computing*, pages 163–170, 2000.

[78] Mathias Kraus, Stefan Feuerriegel, and Asil Oztekin. Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, 281(3):628–641, 2020.

[79] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[80] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, volume 22, pages 1042–1050. Citeseer, 2009.

[81] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Proc. of ICLR '17*, 2017.

[82] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[83] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. of ICML '14*, pages II–1188–II–1196, 2014.

[84] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[85] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. Machine learning in banking risk management: A literature review. *Risks*, 7(1):29, 2019.

[86] Debang Li, Junge Zhang, and Kaiqi Huang. Universal adversarial perturbations against object detection. *Pattern Recognition*, 110:107584, 2021.

[87] Jing Li, Xiaohui Kuang, Shujie Lin, Xu Ma, and Yi Tang. Privacy preservation for machine learning training and classification based on homomorphic encryption schemes. *Information Sciences*, 2020.

[88] Wen-Hui Li, Song Yang, Yan Wang, Dan Song, and Xuan-Ya Li. Multi-level similarity learning for image-text retrieval. *Information Processing & Management*, 58(1):102432, 2021.

[89] Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proc. of ACL '18*, pages 25–30, July 2018.

[90] Hyunki Lim and Dae-Won Kim. Pairwise dependence-based unsupervised feature selection. *Pattern Recognition*, 111:107663, 2021.

[91] F. Limin and M. Enzo. Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC Bioinformatics*, 8(1), Jan 2007.

[92] Kunpeng Liu, Nitish Uplavikar, Wei Jiang, and Yanjie Fu. Privacy-preserving multi-task learning. In *Proc. of ICDM '18*, pages 1128–1133. IEEE, 2018.

[93] Peng Liu, YuanXin Xu, Quan Jiang, Yuwei Tang, Yameng Guo, Li-e Wang, and Xianxian Li. Local differential privacy for social network publishing. *Neurocomputing*, 391:273–279, 2020.

[94] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[95] Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, 2, 2007.

[96] Yuandong Luan and Shaofu Lin. Research on text classification based on cnn and lstm. In *2019 IEEE international conference on artificial intelligence and computer applications (ICAICA)*, pages 352–355. IEEE, 2019.

[97] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proc. of NIPS '17*, pages 4765–4774, 2017.

[98] Yarong Lv. Data privacy protection based on homomorphic encryption. In *Journal of Physics: Conference Series*, volume 2037, page 012129. IOP Publishing, 2021.

[99] Kevin J. Macleod and W. Robertson. A neural algorithm for document clustering. *Information Processing & Management*, 27(4):337–346, 1991. Special Issue: Parallel Processing and Information Retrieval.

[100] Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45:61–68, 2014.

[101] Yury A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2020.

[102] J. Marek, J. Martin, and R. Konrad. Smart metering de-pseudonymization. In *Proc. of ACSAC '11*, pages 227–236. ACM, 2011.

[103] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[104] Ricardo Mendes and João P Vilela. Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017.

[105] B. Michael and Z. Tom. A Face is exposed for AOL searcher no. 4417749. *New York Times*, page A1, 08 2006.

[106] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[107] V. Misra and S. Bhatia. Bernoulli embeddings for graphs. In *Proc. of AAAI '18*, 2018.

[108] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[109] M. Noman, C. Rui, F. Benjamin, and Philip S Y. Differentially private data release for data mining. In *Proc. of KDD '11*, pages 493–501, 2011.

[110] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proc. of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 506–519. Association for Computing Machinery, 2017.

[111] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[112] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[113] Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764, Beijing, China, July 2015. Association for Computational Linguistics.

[114] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. *Advances in neural information processing systems*, 22:1509–1517, 2009.

[115] Mohammad Saidur Rahman, Ibrahim Khalil, Abdulatif Alabdulatif, and Xun Yi. Privacy preserving service selection using fully homomorphic encryption scheme on untrusted cloud service platform. *Knowledge-Based Systems*, 180:104–115, 2019.

[116] Wang Ren, Xin Tong, Jing Du, Na Wang, Shan Cang Li, Geyong Min, Zhiwei Zhao, and Ali Kashif Bashir. Privacy-preserving using homomorphic encryption in mobile iot systems. *Computer Communications*, 165:105–111, 2021.

[117] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *Proc. of ICMLA '15*, pages 896–902. IEEE, 2015.

[118] Sara Rosenthal and Kathleen McKeown. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 763–772, June 2011.

[119] Théo Ryffel, Edouard Dufour-Sans, Romain Gay, Francis Bach, and David Pointcheval. Partially encrypted machine learning using functional encryption. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4519–4530, 2019.

[120] C. Moses S. Similarity estimation techniques from rounding algorithms. In *Proc. of STOC '02*, pages 380–388. ACM, 2002.

[121] Christopher De Sa. Classifying chess positions. Master's thesis, Stanford University, 2012.

[122] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 471–480, 2019.

[123] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. Privacy-preserving record linkage using bloom filters. *BMC medical informatics and decision making*, 9(1):1–11, 2009.

[124] Rainer Schnell and Christian Borgs. Randomized response and balanced bloom filters for privacy preserving record linkage. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 218–224. IEEE, 2016.

[125] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[126] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[127] Procheta Sen and Debasis Ganguly. Towards socially responsible ai: Cognitive bias-aware multi-objective learning. In *Proc. of AAAI '20*, volume 34, pages 2685–2692, 2020.

[128] Shuo Shang, Lisi Chen, Christian S Jensen, Ji-Rong Wen, and Panos Kalnis. Searching trajectories by regions of interest. *IEEE Trans. KDE*, 29(7), 2017.

[129] Shuo Shang, Ruogu Ding, Bo Yuan, Kexin Xie, Kai Zheng, and Panos Kalnis. User oriented trajectory search for trip recommendation. In *Proc. of ICEDT '12*, 2012.

[130] Shuo Shang, Ruogu Ding, Kai Zheng, Christian S. Jensen, Panos Kalnis, and Xiaofang Zhou. Personalized trajectory matching in spatial networks. *VLDB J.*, 23(3):449–468, 2014.

[131] Xiaobo Shen, Weiwei Liu, Ivor Tsang, Fumin Shen, and Quan-Sen Sun. Compressed k-means for large-scale clustering. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[132] Yucheng Shi, Yahong Han, Quanxin Zhang, and Xiaohui Kuang. Adaptive iterative attack towards explainable adversarial robustness. *Pattern Recognition*, 105:107309, 2020.

[133] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[134] Priyanka Singh, Abhishek Singh, Gabriel Cojocaru, Praneeth Vepakomma, and Ramesh Raskar. Ppcontacttracing: A privacy-preserving contact tracing protocol for COVID-19 pandemic. *CoRR*, abs/2008.06648, 2020.

[135] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[136] Jiabao Sun, Jiajie Xu, Rui Zhou, Kai Zheng, and Chengfei Liu. Discovering expert drivers from trajectories. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1332–1335. IEEE, 2018.

[137] Lin Sun, Lan Zhang, and Xiaojun Ye. Randomized bit vector: Privacy-preserving encoding mechanism. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1263–1272, 2018.

[138] Latanya Sweeney. Matching known patients to health records in washington state data. *arXiv preprint arXiv:1307.1370*, 2013.

[139] Latanya Sweeney, Akua Abu, and Julia Winn. Identifying participants in the personal genome project by name (a re-identification experiment). *arXiv preprint arXiv:1304.7605*, 2013.

[140] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[141] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *Proc. of ICLR '14*, 2014.

[142] Jiajian Tang, Zhenfu Cao, Jiachen Shen, and Xiaolei Dong. Lpcp: An efficient privacy-preserving protocol for polynomial calculation based on crt. *Applied Sciences*, 12(6):3117, 2022.

[143] Lu-An Tang, Yu Zheng, Xing Xie, Jing Yuan, Xiao Yu, and Jiawei Han. Retrieving k-nearest neighboring trajectories by a set of point locations. In *International Symposium on Spatial and Temporal Databases*, pages 223–241. Springer, 2011.

[144] Harry Chandra Tanuwidjaja, Rakyong Choi, Seunggeun Baek, and Kwangjo Kim. Privacy-preserving deep learning on machine learning as a service - a comprehensive survey. *IEEE Access*, 8:167425–167447, 2020.

[145] R Thenmozhi, S Shridevi, Sachi Nandan Mohanty, Vicente García-Díaz, Deepak Gupta, Prayag Tiwari, and Mohammad Shorfuzzaman. Attribute-based adaptive homomorphic encryption for big data security. *Big Data*, 2021.

[146] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, 03 2018.

[147] Katherine Turner. Introduction to infectious disease modelling. *Sexually transmitted infections*, 87, 11 2010.

[148] Jaideep Vaidya, Murat Kantarcıoğlu, and Chris Clifton. Privacy-preserving naive bayes classification. *The VLDB Journal*, 17(4):879–898, 2008.

[149] Matthew Veres and Medhat Moussa. Deep learning for intelligent transportation systems: A survey of emerging trends. *IEEE Transactions on Intelligent transportation systems*, 21(8):3152–3168, 2019.

[150] V. S Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *Sigmod Record*, 33:50–57, 2004.

[151] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11:2837–2854, December 2010.

[152] De Wang, Feiping Nie, and Heng Huang. Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 306–321, 2014.

[153] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393, 2014.

[154] Rong Wang, Benjamin CM Fung, Yan Zhu, and Qiang Peng. Differentially private data publishing for arbitrarily partitioned data. *Information Sciences*, 553:247–265, 2021.

[155] Ru Wang, Lin Li, Xiaohui Tao, Xiao Dong, Peipei Wang, and Peiyu Liu. Trio-based collaborative multi-view graph clustering with multiple constraints. *Information Processing & Management*, 58(3):102466, 2021.

[156] Sheng Wang, Zhifeng Bao, J Shane Culpepper, Timos Sellis, Mark Sanderson, and Xiaolin Qin. Answering top-k exemplar trajectory queries. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 597–608. IEEE, 2017.

[157] Yilun Wang, Yu Zheng, and Yexiang Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 25–34, 2014.

[158] Benjamin Weggenmann and Florian Kerschbaum. Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 305–314, 2018.

[159] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18, 2005.

[160] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in neural information processing systems*, 21:4, 2008.

[161] J. William and L. Joram. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[162] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[163] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[164] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:521–528, 2002.

[165] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489. Association for Computational Linguistics, 2016.

[166] Jieping Ye, Zheng Zhao, and Mingrui Wu. Discriminative k-means for clustering. *Advances in neural information processing systems*, 20:1649–1656, 2007.

[167] Xinyang Yi, Constantine Caramanis, and Eric Price. Binary embedding: Fundamental limits and fast algorithm. In *International Conference on Machine Learning*, pages 2162–2170. PMLR, 2015.

[168] Kaiwei Zeng, Munan Ning, Yaohua Wang, and Yang Guo. Hierarchical clustering with hard-batch triplet loss for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13657–13665, 2020.

[169] Lingchen Zhao, Qian Wang, Cong Wang, Qi Li, Chao Shen, and Bo Feng. Veriml: Enabling integrity assurances and fair payments for machine learning as a service. *IEEE Trans. Parallel Distributed Syst.*, 32(10):2524–2540, 2021.

[170] Wenzhao Zheng, Jiwen Lu, and Jie Zhou. Deep metric learning via adaptive learnable assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2960–2969, 2020.

[171] Peng Zhou, Liang Du, Xuejun Li, Yi-Dong Shen, and Yuhua Qian. Unsupervised feature selection with adaptive multiple graph learning. *Pattern Recognition*, 105:107375, 2020.