Project Final Report

Privacy-Preservation and Implications of Machine-Learning

Michael Rist and Soudeh Nilforoushan
mrist@uwo.ca, snilforo@uwo.ca

The University of Western Ontario
Department of Computer Science

1 Background and Significance:

The collection of digital information by different organizations has created tremendous opportunities for knowledge-based decision making. Add to this an increasing demand for exchanging data among different parties, and the result is rising concerns over privacy. In response, various techniques have emerged to restrict sensitive publishable data and preserve individual privacy, while also maintaining the usefulness of the published data. We call this privacy-preserving data publishing (PPDP)[1]. This approach also applies in privacy preserving machine learning(PPML). Aspects of PPDP and PPML, can be applied more directly with respect to databases (ie. anonymization, data redaction and some encryption). Furthermore, there exists a number of commercial database tools that can be used to employ some of these 'baseline' aspects of PPDP/PPML. However, preserving privacy while maintain the usefulness of the published data often requires techniques beyond those provided by commercial tools.

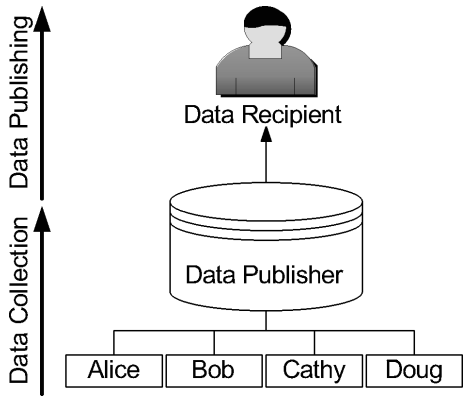


Figure 1: Data Collection and Data Publishing [1]

1.1 Privacy Preservation

In a typical scenario of PPDP, there is two phases: In the data collection phase, a data publisher receives data from record owners (i.e. hospital patients); then in the data publishing phase, a data recipient collects the data from the data publisher. This process is described in Figure 1. The data recipient could be an attacker, and even in scenarios where the data recipient is 'trusted', the guarantee of 'trustworthiness' cannot always be transitive to the data recipient's staff[1]. So, before a data publisher sends data to data recipients, some privacy-preserving techniques should be applied. Where the data directly identifies record owners or contains identifying attributes (known as Quasi Identifiers (QID)), data publishers should anonymize the records and not publish the Quasi Identifiers (QID) of data owners.

Privacy attacks and corresponding privacy preservation models fall into one of two categories. First, there is are attacks associated with record linkage, attribute linkage and table linkage. Here, we assume that the attacker knows the QID of the victim. The second category is associated largely with probabilistic attacks and the corresponding privacy preservation methods aim for the achievement of uninformative principle. Here the data publisher seeks to publish tables that provide the attacker with little more than background knowledge[1].

PPDP (with its various tools and methods) is often applied as a generalist approach to privacy preservation, where downstream application/use of the data may not be known at the time of publishing[2]. The focus of PPDP

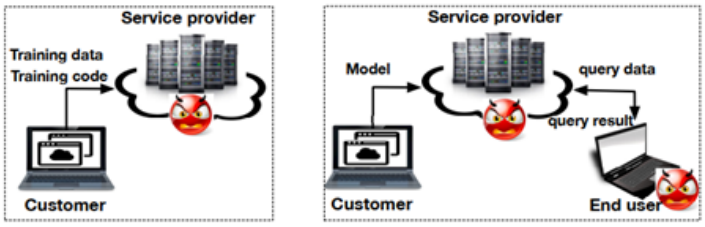
is on data publishing, with the expectation that standard data mining techniques can be applied on the resultant data. Furthermore, PPDP seeks primarily to 'anonymize' data, protecting record-owner identities, but often with an emphasis on preserving data truthfulness at the record level [2, 1]. With the emergence and increased use of machine-learning (ML), a new suite of threats now exist that present a challenge to PPDP.

1.2 Machine Learning (ML)

Machine-learning (ML) has become increasing popular and ubiquitous in our society, with applications ranging from product recommendation to spam/malware filters and online fraud detection systems. Often these applications use private data to extract patterns and build models [3, 4]. Alongside standard insider / outsider threats associated with the centralized storage of data, the use of machine-learning presents additional challenges to privacy. In some instances, machine-learning has the potential to reveal sensitive or individually identifying information on previously anonymized data. Al-Rubaie et al [3] discusses the risk of De-Anonymization using a Netflix example, wherein Netflix released an anonymized dataset for use in an open competition, with the aim of building a better recommender system. Using this published dataset, researchers were able to use publicly available background knowledge to identify some individual record identities from the Netflix dataset. Al-Rubaie et al [3] also discusses several other lines of ML-associated threats to privacy including reconstruction, model-inversion and membership inference attacks. Adding to this, with the continuing growth and evolution of cloud services, Machine Learning-as-a-Service (MLaaS) has become increasing popular.

1.3 Machine Learning-as-a-Service (MLaaS)

ML, specifically the use of Deep Neural Networks, is yielding remarkable results across a number of domains [4]. However, the increasing need for computation and storage resources (for running ML) has many companies turning to cloud-based MLaaS. MLaaS allows customers to outsource model training and / or model serving tasks to cloud service providers, but in turn requires the customers to place trained models and / or data in the cloud [5, 6]. In the model serving paradigm (figure 2 (b)), a customer uploads a pre-trained model to the cloud service provider. The cloud service provider stores the model in the cloud, making query APIs available to the customer to use the model for prediction or classification. The model training services paradigm (figure 2 (a)) has the customer providing the training dataset (and sometimes the the ML algorithms) to the cloud service provider, who then sets up the environment, allocates resources and runs the model training task. In some cases, this model is then stored in the cloud and made available to the customer through the aforementioned model serving paradigm. Within the context of MLaaS, there are three principle scenarios related to privacy concerns (see figure 2), which are discussed by Zhang et al. [5]. In the first scenario, the customer uploads a dataset to the ML service provider, giving access to the full dataset. If the service provider is malicious, they can steal the sensitive data and / or embed the data into the model for leakage. The second scenario relates to the use of a model serving service, wherein the customer uploads a pre-trained model to the cloud (setting up endpoints for users). If ML service provider is malicious, even though they don't have access to the full dataset, they can still extract sensitive info about the dataset from model parameters. Finally, the third scenario is a case where the ML service provider is trusted, however, a malicious remote user (even with just black box access to model output) is still able to extract information about the data using model queries and varying inputs. Through these scenarios, Zhang et al. [5] draws out specific attack models. In addition to membership inference and model inversion attacks (also outlined by Al-Rubaie et al [3]), Zhang et al [5] highlights ML model memorization and model classification attacks.



(a) Model training (b) Model serving

Figure 2: Attack Settings and MLaaS Service Modes [5]

1.4 Privacy Preservation in Machine Learning

In the light of these evolving concerns, **privacy-preserving machine learning** (PPML) has emerged with a focus on providing more specific techniques for addressing some of these vulnerabilities. PPML is largely concerned with tools and techniques that can be applied to maintain anonymity and privacy of data that is subject to the application of machine learning. One of the challenges associated with PPML however, is that it is often tied to specific ML algorithms, and thus may be difficult to generally apply as a data publisher.

1.5 Significance

The effectiveness of PPDP/PPML techniques in maintaining anonymity and safeguarding privacy in the face of downstream ML applications is significant, such that as ML becomes more prevalent, so do the potential risks to an individual's assumed data privacy and anonymity.

Paper Organization

The remainder of this paper is organized as follows. Section 2 presents the Goals and Objectives of this paper. Section 3 discusses the papers approach to accomplishing the goals laid out in section 2. Section 4 outlines the challenges faced in the development of the paper. Section 5 discusses threats and attack models associated with the downstream application of machine learning on data or the use of Machine Learning-as-a-Service. Section 6 discusses available database tools that can contribute to privacy preservation. Section 7 outlines and details solutions (privacy models) that can be used against the ML threats discussed in section 5. Section 8 presents results and discusses the effectiveness of the solutions in the context of their performance and tandem use. Section 9 concludes the paper areas of future work/interest.

2 Goals and Objective:

/ The goal of this paper will be to identify and compare a number of PPDP and PPML techniques, evaluating their relative effectiveness against emerging ML-associated threats.

In order to accomplish this goal, this paper will consider and discuss several known privacy-preservation threats associated with the application of ML on published data. PPML-specific techniques will then also be discussed through a similar lens, along with an evaluation of their relative effectiveness. Finally, this paper will summarize with a comparative analysis of the discussed techniques, laying out conclusions and highlighting related areas for future study.

3 Approach:

Our initial topic search revealed a number of source papers, providing us with an initial knowledge foundation of PPDP, PPML and the possible implications of ML on privacy. In order to expand upon this, we will be considering additional sources in order to reveal and consolidate the current state-of-the-art for ML-associated privacy threats, as well as relevant PPDP/ PPML techniques. Searches will be conducted using Google Scholar, UWO's OMNI Academic Search tool, as well as the IEEE Xplore, ACM Digital, and SpringerLink Databases. We will also use chain searching techniques (backward / forward reference searching) to identify additional relevant papers. Based on these sources, we will select relevant privacy techniques that can be mapped to known ML-associated attacks.

In order to compare and evaluate the 'relative effectiveness' of the discussed privacy models and techniques, it is our intent to map these individual techniques to various ML-associated attack models, indicating for which attack model(s) the individual technique is considered effective (from the literature). The resultant matrix is expected to summarize and reveal the relative effectiveness of individual techniques across the spectrum of ML-associated threats. Hussien et al [1] employed a similar summary technique to map various PPDP privacy models to a number of categorical attack models.

4 Challenges:

Comparing and evaluating the 'relative effectiveness' of various PPDP and/or PPML associated techniques may present some challenges. In the approach section above, we discuss how we intend to map various PPDP and PPML techniques to ML-associated attack models. One possible challenge will be the availability of source material capable

of revealing certain PPDP techniques' effectiveness against ML-associated attacks. Much of the available literature concerning PPDP techniques is concerned with general effectiveness against more traditional attack models, such as record/attribute/table linkage attacks and probabilistic attacks[1, 2]. In such cases there may be some overlap between some of these traditional attack models and more ML-specific attacks, for example membership inference attacks. Should cases arise where source information is not sufficient to map a technique to an ML-specific attack model, possible remedies may include exclusion of the technique in the study, partial inclusion, or in some cases deeper analysis of the techniques' algorithm may reveal information allowing us to map it to a ML-specific attack.

Although PPML has been around for a while, only recently have we seen it truly break through as an area of intensive research. As a result, there is still a somewhat limited amount of available literature compared to PPDP. We don't expect this to present a major challenge to our study, as we are focused on the current state-of-the-art. It does however lead to an interesting challenge to the study of PPML in general. The novelty of current PPML research, along with new ideas in Machine Learning constantly being proposed, creates additional challenges in PPML due to the fact that existing techniques are tied to specific ML algorithms. Therefore, new techniques in PPML need to be constantly proposed according to new algorithms of Machine Learning[3]. For example, in response to the increased use of ML-as-a-service, Ping Li et al [7] discusses a novel approach using a public key encryption with a double decryption algorithm (DD-PKE) for cloud data from different data providers.

5 ML Threats

Consider our earlier discussion of the roles in PPDP / PPML (input, computation and results parties). If these roles are assumed by more than one entity, then a possible attack surface is created and privacy-preserving technologies / techniques are required [3, 4]. In the following attack models, it is assumed that the input party (data owner) and the computation party are always separate entities. In some attack models, there may also exists an end user (results party). In these cases, it is specified if they are a different entity than the input or computation party. It should also be noted that in some cases we use the term 'service provider' interchangeably with or instead of computation party. This is because in some attacks, the computation party may be unknown to the data owner / input party. Whereas, in the case of MLaaS, the computation party is known (although possibly still untrusted). In cases where we refer to MLaaS 'service provider', this is in relation to the MLaaS service modes in figure 2.

5.1 ML/MLaaS Attack Models

Data in the Clear / ML Model Memorization Attacks

This is the simplest attack model scenario. In this case the data is stored in its raw form and is not encrypted, transformed or anonymized in way. This leaves the data highly vulnerable to attacks from inside and outside the computation party (with whom the data now resides). [3]

Another form of this attack type is referred to as **ML model memorization attack**. This is largely related to scenarios where the data owner outsources model training to a MLaaS provider. Zhang et al [5] outlines this type of attack, wherein the service provider (computation party) is given full access to the dataset. The provider, or a malicious actor inside the provider, can simply steal the data and / or can encode the sensitive data into the trained model using various methods (ie. LSB encoding, correlated value encoding, sign encoding or by abusing model capacity). In the second part of this attack, provided the malicious party has access to the model, they can retrieve the sensitive information about the dataset. In the case that the malicious party has 'white-box' access to the model (through the model serving MLaaS paradigm), they can simply retrieve the sensitive information about the dataset from the model parameters [5]. If they only have 'black-box' access to the model (as an end user with API query access), they can still retrieve the sensitive information about the dataset from model outputs[5].

Reconstruction Attacks

In this case, the computation party does not have full access to the dataset. Rather they have 'white-box' access to the trained model, including feature vectors [3]. This model is similar to and overlaps with the second stage of the **ML model memorization attack** [5] outlined above. In reconstruction attacks, the malicious party aims to reconstruct the raw dataset by using information from the feature vectors. In [3], attacks of this nature are often made possible if the feature vectors from the training phase are not deleted thereafter and are stored in the model. According to [3], SVM and k-nearest neighbors (kNNs) are examples of ML models that natively store feature vectors in the model itself.

Model Inversion Attacks

Now consider ML models that do not store feature vectors in the model itself. Thus, model access is limited. One type of access is where the computation party/ service provider is simply storing and serving a pre-trained model (basic white-box). The other form of access is that of a results party / end user with API query access to the ML model for prediction (black-box). In this attack paradigm, the malicious party is working to recreate the feature vectors (as close as possible) to those originally used to create the model [3]. In the case that the attacker has basic white-box access (ie. service provider), and is able to identify the model (algorithm, topology and parameters), they are able to reveal information from the training dataset. Zhang et al [5] provides an example, wherein the attacker (service provider) was able to use model parameters as features to reveal sensitive properties about the training dataset. Where the attacker only has black-box access to the model output, its is possible for them to send crafted samples to the model and use the output (predictions) to learn about the dataset. In some model inversion attacks, the attacker uses the output confidence score and the optimized sample whose confidence score is most close to 1 for one specific class [3, 5]. Zhang et al [5] points out that, in this case, exposure is greatest where a class is representative of a single individual. However, in most cases model inversion attacks do not reconstruct individual samples. Rather, the recreated feature vectors that result can instead be used as an input for reconstruction attacks.

Membership Inference Attacks

In membership inference attacks [3, 5], the attacker (results party) is armed with some knowledge of a target sample. Using black-box access to the ML model, the attacker seeks to 'infer' if the sample is a member of the training dataset that was used to train the model. Truex et al [8] provides an example to demonstrate the motivation behind a membership inference attack. In the example, an attacker seeks to determine if a patient is part of a training dataset that was used to train a model that predicts cancer-related health outcomes. By doing so, the attacker is able to determine that patient was part of the dataset and uses this to act against the patients interests. Similar to model inversion above, the attacker uses output confidence scores from predictions on optimized samples as a training dataset to build an inference attack model [3, 5, 8]. This, in turn, is used to predict if the target sample is a member of the original ML model's training dataset [3, 5, 8].

De-Anonymization

In order to anonymize sensitive data, the removal or encryption of personal identifiers (name, ID, address) is common practice. In spite of these efforts, attackers are still able to de-anonymize or re-identify individual records or information using auxiliary information [3, 9]. Data often passes through multiple sources, de-anonymization techniques utilize this availability of (public) information to cross-reference the anonymized data, revealing identity. As discussed in section 1.2, Al-Rubaie et al [3] uses a well known Netflix example to illustrate de-anonymization. The authors example shows that without any record identifiers on the Netflix data, an attacker armed with only a little information about a subscriber can in fact identify their record in the dataset.

Model Classification Attacks

In this model, the attacker's aim is the learn the attributes present in a training dataset. Zhang et al [5] describes a scenario in which the attacker (Service provider in a model serving paradigm) with white-box access is able to infer the attributes of the dataset from the model parameters. The authors describe a process such that, similar to membership inference attacks, the attacker is able to build 'shadow' attack models (both with and without the target attribute). The parameters of these models are then extracted as features and fed into a classifier, which is used to predict if the property exists in the original training set.

6 Database Tools for Privacy Preservation

The focus of this paper is on privacy preservation for data that may accessed by downstream ML applications, not data 'at rest' in a database. However, there is still some basis that the discussion of privacy preservation should start at the database. Databases remain a key source for data driven applications and often house some of the most critical and sensitive data. As seen in section 5, even when data has been anonymized and / or encrypted, there exist attack models capable of overcoming these privacy efforts. Regardless of this, basic anonymization and, in some cases, encryption form a foundational step in any PPDP/PPML approach. To this end, there exist a number of commercially available database tools that can contribute to this essential step in PPDP/PPML.

One such tool set is IBM Guardium. In addition to general database security features such as access management and monitoring, Guardium also provides tools for the automated discovery of sensitive data, data redaction, masking and encryption [10]. These tools work largely at the database level and have the capacity to work for 'on-premise' systems or across hybrid multi-cloud environments. Similarly, Oracle Advanced Security (and to a lesser extent Oracle AVDF [10]) also offer tools capable of achieving some privacy preservation, including data encryption and data redaction. Thales VDS Platform, another database security tool set, offers tools geared toward database encryption and data protection [10]. Finally, Always Encrypted (AE) is another tool made exclusively for MS SQL and Azure SQL databases [10]. AE has a more limited scope than the other previously discussed tools, working solely to encrypt sensitive information. These tools represent only a cross-section of the available database security tools which are capable of contributing towards the goals of PPDP/PPML.

In the following section (PPML Solutions / Privacy Models), techniques beyond basic database encryption and anonymization are discussed. Such solutions are required to address the more complex attack models discussed in Section 5. The solutions discussed will, in most cases, build upon this foundation of anonymized and / or encrypted data that has been addressed through these database tools. In most cases these solutions do not integrate with databases, rather they are applied on data extracted from the database. One exception to this is with order-preserving encryption (a cryptographic approach), which can be applied directly to a database.

7 PPML Solutions / Privacy Models

There are two main categories for privacy-preserving machine learning and we discussed in follows:

7.1 Perturbation Approaches

The main idea of perturbation is to add some noise to the input or output of the model in the training phase. There are three categories of privacy model for this approach:

Differential Privacy (DP)

This approach minimizes the possibility of recognizing each individuals' record in the data set [6]. In other words, we say an algorithms is deferentially private if we look at the output and one cannot tell whether any individual's data was included in the original data set or not. Furthermore, when applied its behaviour hardly changes when a singe individual joins or leaves the data set. Therefore, regardless of how eccentric any single individual's details are, and regardless of the details of anyone else in the data set, the guarantee of differential privacy still holds. This gives a formal guarantee that individual-level information about participants in the database is not leaked.

ε is a positive real number and M is a randomized algorithm that takes data set D_1 and D_2 as input and P is the probability. Algorithm M is said to provide ε-differential privacy if, for all data sets D_1 and D_2 that different on a single element, and all subsets S of M:

$$Pr[M(D_1) \in S] \leq exp(\varepsilon) Pr[M(D_2) \in S]. \quad (1)$$

Differential privacy protects the data when we have multiple input sides. On the other hand, an attacker cannot de-anonymize differential privacy since this method is immune to post-processing. There are three different phases for differential privacy to add noise to the data set

- input noise:noise is added to the input and after the computation on the noisy input the out put would be differently private.
- algorithm noise: noise is added to intermediate values in iterative algorithms.
- noisy output: At this stage we run the non-private-learning algorithm and then the noise is added to the generated model.[6, 2].

Local Different Privacy (LDP)

"When the input parties do not have enough information to train an ML model, it might be better to utilize approaches that rely on LDP." [2].The assumption here is that the data collector is untrusted so each party adds noise to their data set and publish the perturbed data to the untrusted server.[11]

Dimension Reduction (DR)

"This approach modify the source data by projecting them to a lower dimensional hyperplane of smaller dimension."[3] Retrieving the original data from reduced dimension is impossible because the number of unknown variables are more than the number of equations. Several techniques have been introduced to reduce the dimension of a matrix. There is a trade-off, as much as we reduce the dimension, we also increase the privacy but we lose the utility. Some techniques have been introduced that combine DR and LDP, because the approximation of original data can retrieved from DR when used by itself[3, 2].

7.2 Cryptography approaches

There are four different categories for Cryptography approaches.

Homomorphic Encryption

This approach is the most common method in PPML. Homomorphic encryption uses addition and multiplication with the encrypted text and retrieves an encrypted result. The following equation shows this approach[6]

$$\begin{cases} D(E(m_1) \oplus E(m_2)) = m_1 + m_2 \\ D(E(m_1) \otimes E(m_2)) = m_1 * m_2 \end{cases} \tag{2}$$

m1 and m2 are plain text, D is a decryption function and E is a encrypted function. There is performance disadvantage for Homomorphic Encryption, such that this approach is slow and needs increased memory, however it has good accuracy. So, some machine learning algorithms cannot use this approaches since machine learning techniques also have high memory requirements themselves[12, 6].

Garbled Circuits

This approach allows two parties that do not trust each other to jointly evaluate their secret data without involving a trusted third party. The disadvantage of this approach is that this technique needs to be implemented for each machine learning techniques, so it increases complexity and run time.[6]

Secure Processors

"This approach protects sensitive data from modification or unauthorized access."[6]. In this technique we have multiple data owners to perform a machine learning task, with the computational party running the ML task at the data center. In this system each data owner independently establishes a secure channel with the code and data, then authenticates themselves. This verifies the integrity of he ML code in the cloud and uploads the private data to the enclave[3]. This approach has a drawback, the processor only protects the code, not input or output. So, we have to apply other techniques for this problem.[6]

Order-Preserving Encryption

"In this method the encryption function preserves the numerical order of plaintext. So, it maintains the confidentially of information. Order-Preserving is faster than homomorphic. This approach has several features:

- the results of encryption are accurate.
- It can combine easily with existing databases.
- when we add or delete a column or a value it does not need to change the encryption of other values, so it is flexible.[6]

8 Results

Figure 3 maps the various ML specific attack models (threats) to the various privacy models (solutions) discussed in Section 6. A check mark in the table indicates that the privacy model is effective against the corresponding attack model. In looking at the perturbation models, differential privacy (DP) and local differential privacy (LDP) work effectively to counter 'inference' related attacks. DP has also shown promise in application for the prevention of model inversion attacks [13]. Dimension reduction (DR) on the other is effective on data / model memorization and

	Privacy Model	Attack Model					
		Data in the Clear/ ML Model Mem.	Reconstruction Attack	Model Inversion Attack	Membership Inference Attack	De-Anonymization	Model Classification Attack
Perturbation	Differential Privacy			✓	✓	✓	✓
	Local Differential Privacy				✓		✓
	Dimension Reduction	✓*	✓			✓*	
Cryptography	Homomorphic	✓	✓	✓			
	Garbled Circuits	✓	✓				
	Secure Processors	✓	✓				
	Order-Preserving Encryption	✓**					

✓ indicates the privacy model is effective

✓* effective when data is anonymized.

✓** can only be used with decision tree-based ML algorithms

Figure 3: Model Mapping: Attack Model to Privacy Model

reconstruction attacks. From the cryptographic models, it can be seen that homomorphic encryption is effective in preserving privacy against model memorization, reconstruction and model inversion attacks. Similarly, garbled circuits can also be used effectively to counter model memorization and reconstruction (model input and output) attacks. Theoretically, secure processors should also guard against model memorization and reconstruction attacks, as the computations all take place in secure enclaves (provided no sensitive data is stored with the computation party outside this enclave). Finally, order-preserving encryption should theoretically provide cover against various data in the clear and model memorization attacks. Lisin et al [6] theorizes that, although this method limits the set of possible operations when used as a transformation of data for input into ML algorithms, it can still be used with decision tree based ML (which uses these same support operations). Whats more, this method allows the use of databases. An example used by Lisin et al [6] is such that a hospital could provide its database to a third party service provider and, provided it has been encrypted with order-preserving encryption, the service provider can analyze the data (with available operations) without compromising confidentiality. An additonal benefit associated with this privacy model is higher speed performance.

As can be seen in figure 3, there is no 'catch-all' privacy model that can counter all of the various threats. Rather combinations are required to create a holistic approach towards privacy preservation. One such combination is the use of differential privacy together with dimension reduction. As can be seen in figure 3, the compliment of these two privacy models has the capability of preserving privacy across the entire spectrum of threats / attack models. Supporting this observation, Al-Rubaie et al [3] discusses previous work in which DP and DR are used together to completely obfuscate the data. Based on characteristics of perturbation methods and these specific privacy models, we would expect this combination to perform better in terms of speed and memory requirements, but less so in terms of accuracy. Other combinations involve both perturbation and cryptographic approaches being applied together. One such combination is the use of homomorphic encryption together with differential privacy [13]. This combination also counters the full spectrum of attack models. Thaines [13], supports this observation in claiming this combination covers off the 'four pillars of privacy preservation for AI', which includes training data, input data, model and output data privacy. The use of homomorphic encryption in this combination would be expected to increase computational resource requirements. Thaines also also highlights another similar combination, which uses secure multiparty computation (a form of garbled circuits) along with differential privacy. This also supported by the mapping seen in figure 3.

As the results show (figure 3), privacy models hold varying degrees of effectiveness and when used in combination can compliment each other to subvert a range of attacks or, in some cases, work in concert to counter a single attack model. There are, however, also best practices that can be employed along side these privacy models to reduce potential attack surfaces. Firstly, to avoid data theft (ie. data in the clear attacks) and / or the encoding of sensitive data into ML models, sensitive data in its raw form should never be publish. This opens the data owner to the largest possible attack surface. Second, attempts to anonymize data by removing personal identifiers can also remove some of these simpler attack threats. As seen in section 6, database security tools exist that can automate and facilitate these practices. Regardless, in a scenario with a possibly **untrustworthy** computation party, publishing any data (even anonymized or encrypted data) can still leave a significant attack surface. By withholding access by any computation party or service provider to the dataset, the attack surface can be significantly reduced. This, however, means that the data owner's outsourcing options are limited to the model serving paradigm only. Also, within the

context of a model serving paradigm, careful selection of the ML models (avoiding those which store feature vectors), can further help in preventing reconstruction attacks. Finally, its important to see that attacks are in some cases used in combination and can lead to a cascade of privacy failures. Attackers with simple black-box access can use model inversion attacks to reconstruct feature vectors, which in turn can be used to facilitate reconstruction attacks. Employing a balance of these best practices, along with appropriate privacy model combinations (such as those mentioned above), can reduce attack surface exposure to an acceptable level.

9 Conclusion

With the development of Machine Learning, the risk of stealing data by an adversary has been increased. In this paper we mainly focus of different kinds of ML-based attacks and provide possible solutions. Downstream ML threats, including those framed by the MLaaS service modes (figure 2), generally involve a form of membership inference, an attempt to recreate the original dataset or to reveal some sensitive information about the original dataset. While some attacks require full access to the original dataset, others may only need white-box access to the ML model and its parameters. In some cases, attacks can even reveal sensitive information with only black-box access. The range of methods and the sophisticated nature of these different attacks make them difficult to counter with any single counter method. There are two main categories for solutions of PPML: the perturbation approach, which has a higher speed; and the cryptographic method, which is more accurate. Each privacy model within these two categories has advantages. Differential privacy is better for the training phase, local differential privacy works well at the data collection phase and dimension reduction algorithms perform well at the training phase. Homomorphic encryption is a popular method and works well for all machine learning techniques, but its slow and has higher memory requirements. Garbled circuits performs well when we have more than one model but, because we have to customize it's implementation for each ML technique, this approach is complicated. Secure processors protect the data from theft and is fast relative to other cryptographic methods, however its still not commonly employed in PPML. Order-preserving encryption is interesting, as it can be employed on databases, however its only relevant for decision tree based ML algorithms. Privacy preservation is a challenging area and the solution space is still somewhat thin. Solutions also typically require a trade off between accuracy and speed. As a result, there exists a multitude of research opportunities.

9.0.1 Future Work

One of the interesting areas for future work would include additional research on the application of order preserving encryption within the context of downstream ML applications. In [6], the use of order-preserving encryption with ML is referenced theoretically. Better performance characteristics (relative to homomorphic encryption), along with its ability to work with relational databases, makes this a potentially unique tool.

Secure processors are still not commonly used in PPML, but show strong promise from a performance perspective. Also, because they are not broadly used in this context, their may in fact prove to have broader effectiveness against various attacks than is theorized in our results discussion. Future work could include an implementation that attempts to use secure processors within context of the two MLaaS service modes, then tests their effectiveness against the various attack models.

References

[1] Benjamin Fung, ke Wang, Rui Chen, and Philip Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42, 06 2010.

[2] Abou-El-Ela Hussien, Nermin Hamza, and Hesham Hefny. Attacks on anonymization-based privacy-preserving: A survey for data mining and data publishing. *Journal of Information Security*, 04:101–112, 01 2013.

[3] M. Al-Rubaie and J. M. Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security Privacy*, 17(2):49–58, 2019.

[4] Ehsan Hesamifard, Daniel Takabi, Mehdi Ghasemi, and Rebecca Wright. Privacy-preserving machine learning as a service. *Proceedings on Privacy Enhancing Technologies*, 2018:123–142, 06 2018.

[5] Tianwei Zhang, Zecheng He, and Ruby B Lee. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860*, 2018.

[6] N. Lisin and S. Zapechnikov. Methods and approaches for privacy-preserving machine learning. In Sergey Yu. Misyurin, Vigen Arakelian, and Arutyun I. Avetisyan, editors, *Advanced Technologies in Robotics and Intelligent Systems*, pages 141–148, Cham, 2020. Springer International Publishing.

[7] Ping Li, Tong Li, Heng Ye, Jin Li, Xiaofeng Chen, and Yang Xiang. Privacy-preserving machine learning with multiple data providers. *Future Generation Computer Systems*, 87:341–350, 2018.

[8] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, pages 1–1, 2019.

[9] AI Network. A brief survey on privacy preserving machine learning techniques. *Available: https://medium.com/ai-network/a-brief-survey-on-privacy-preserving-machine-learning-techniques-b7883b5e6c33*, 2018.

[10] Drew Robb. Top database security tools for 2021. *Available: https://www.esecurityplanet.com/products/database-security-tools/*, 2021.

[11] Teng Wang, Xuefeng Zhang, Jingyu Feng, and Xinyu Yang. A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors*, 20(24):7030, Dec 2020.

[12] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie. Private machine learning classification based on fully homomorphic encryption. *IEEE Transactions on Emerging Topics in Computing*, 8(2):352–364, 2020.

[13] Patricia Thaine. Perfectly privacy-preserving ai: What is it and how do we achieve it? *Available: https://towardsdatascience.com/perfectly-privacy-preserving-ai-c14698f322f5*, 2020.