Data Privacy Protection based on Feature Dilution in Cloud Services

1st Feng Wu
School of Software Engineering
Yunnan University
Yunnan, China
gzwf@mail.ynu.edu.cn

2nd Lei Cui
School of Information Technology
Deakin University
Sydney, Australia
cuil@deakin.edu.au

3rd Jianan Feng
School of Software Engineering
Yunnan University
Yunnan, China
fjn97@mail.ynu.edu.cn

4th Liwen Wu
School of Software Engineering
Yunnan University
Yunnan, China
wulw@mail.ynu.edu.cn

5th Saowen Yao
School of Software Engineering
Yunnan University
Yunnan, China
yaosw@mail.ynu.edu.cn

6th Shui Yu
School of Computer Science
University of Technology Sydney
Sydney, Australia
shui.yu@uts.edu.au

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685603

***Abstract*—Machine learning as a service (MLaaS) brings many benefits to people's daily life. However, the service mode of MLaaS will increase the risk of users' privacy leakage. Existing works focusing on privacy-preserving based on encryption, differential privacy, and distributed framework require high computing resources or cannot be applied in MLaaS. In this paper, we propose feature dilution (FD), a noise-based desensitization algorithm to remove sensitive information in raw data. In particular, FD continuously adds raw data features to the random noise until it meets the minimum amount for an effective query, and we call this noise weak-feature noise (WFN). By fine-tuning the MLaaS architecture, we have realized that users can utilize WFN to get normal services without exposing their local private data. Meanwhile, noise addition technology is introduced by us to reduce the risk of privacy leakage caused by "weak features". Extensive experiments have demonstrated that users can use FD to obtain effective services without exposing their private data. Finally, we conducted practical tests on weak-feature noises and found that these noises are difficult to use by malicious service providers.**

***Index Terms*—MLaaS, privacy-preserving, desensitization, deep neural networks, noise**

I. Introduction

Machine learning as a service (MLaaS) allows machine learning technology to improve people's quality of life [11] effectively. It enables small and medium-sized enterprises and individuals who do not have sufficient data and computing resources to enjoy high-quality artificial intelligence services. However, as a price, these customers must share their private data with service providers. This service mode will continue for a long time. Therefore, the long-term mandatory sharing of private data is undoubtedly a severe violation of individual privacy. For example, a recent study has demonstrated that machine learning-based healthcare services can expose patients' genetic markers [12]. In general, MLaaS can bring significant benefits to both customers and providers, but privacy issues hinder its further development.

Many methods focusing on privacy-preserving in MLaaS have been proposed, and encryption-based technology is the most studied. In particular, Gilad et al. [4] demonstrated that fully homomorphic encryption can be applied to neural networks, enabling the neural network to make encrypted predictions and return the encrypted results. Then, these encrypted predictions can be sent back to the key owner for decryption. Consequently, cloud services will not obtain any information about customers' privacy. Jiang et al. [7] proposed a method named E2DM, which can encrypt multiple matrices into a single ciphertext. E2DM extends some basic matrix operations, such as rectangular multiplication and transpose for advanced operations. However, the time-consuming encryption process limits almost all encryption-based privacy protection strategies, and encrypted data is difficult to provide real-time services.

Recently, federated learning (FL) [15], a novel distributed learning paradigm, has become prominent in addressing privacy issues. It enables data owners to collaboratively train a model without sharing their raw data with the server. However, the FL framework cannot be applied in MLaaS directly. Since in the MLaaS scenario, customers cannot participate in the training phase of the model. To the best of our knowledge, there is no existing solution supporting customers to keep their private data locally while obtaining services normally in MLaaS.

In this paper, we propose feature dilution (FD), the first privacy-preserving scheme that supports customers keeping their private data locally when using the MLaaS services. We have fine-tuned the traditional MLaaS architecture. Specifically, the service provider needs to provide customers with an additional desensitization tool. Customers use the tool to desensitize their private data offline to obtain weak-feature noise (WFN) and finally use WFN to obtain the service, as

This research was funded by the National Natural Science Foundation of China (No. 61863036).

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

shown in Fig. 1. WFN is formed by adding the non-robust features [6] of the private data bit by bit to the random noise until it can satisfy an effective query. Compared to existing protection methods, FD does not require frequent interaction between the client and the cloud, thus it can save a lot of calculation and transmission costs. We verify the proposed method on ImageNet, conduct security analysis, and demonstrate how the noise in FD will affect the service quality.

In summary, the contributions of this paper are summarized as follows.

1. We propose an scheme that clients can obtain services normally without uploading private data to the cloud. As we know, this is the first method that does not require users to upload local data in the MLaaS scenario to obtain services normally.

2. We design a noise-based desensitization algorithm that can remove the most sensitive information while reserving limited diluted features. The developed method will not affect service quality.

3. We analyze the security of weak-feature noise (WFN). Analysis shows that it is tough for dishonest service providers to obtain clients' privacy through WFN and use it for secondary use.

II. Related Work

Andrew et al. [1] morphs the data locally to make people unrecognizable and then submit the morph matrix and data to the server. A particular neural network on the server can provide clients with correct feedback through the data uploaded. However, the morphed data can still be utilized by the dishonest server. Hesamifard et al. [5] build a scheme called CryptoDL to ensure that the data uploaded by clients can be efficiently calculated under encryption. Unlike Cryptonets [4], CryptoDL is more efficient in execution, but it is also limited by computation and communication overhead. Ma et al. [10] block data interaction between clients and service providers by outsourcing. Although it is effective, it requires a trusted third party.

Bost et al. [2] construct a security classification protocol of hyperplane decision-making, naive bayes, and decision tree based on comparison blocks. Tramer et al. [13] use Trusted Execution Environments (TEEs), which isolate the computation process from untrusted software. Li et al. [8] propose a secure online classification service outsourcing solution, but clients will involve frequent interactions when they initiate classification queries.

Although these methods can well protect customers' privacy, they either inevitably generate a lot of computing and transmission overhead or must have a trusted environment or a third-party guarantee.

III. Proposed Method

In this section, we present precisely the problem formulate and specific algorithms.

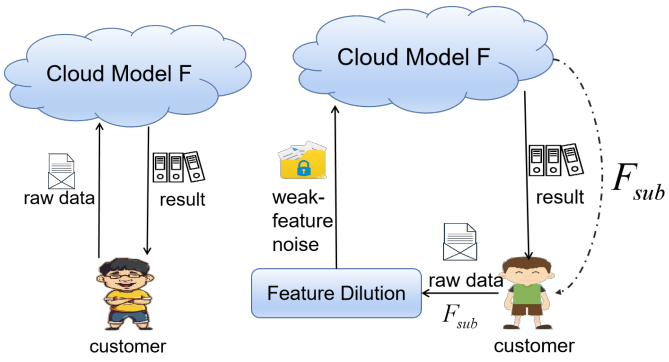


Fig. 1: Left: the traditional MLaaS service mode. Right: the new service scheme, where F_{sub} denotes the desensitization tool that help customers protect their privacy.

A. Problem Formulate

We consider business scenarios such as a cloud model in MLaaS. Given a fixed CNN model F, we specify $L = (l_1, l_2, l_3, ..., l_n)$ as its model structure, where l_j represents the structure of the j-th layer, and F_{sub} means a substructure of F (e.g. a model can be truncated from a specific middle layer to form a new model F_{sub}), the corresponding structure of F_{sub} is$L_{sub} = (l_1, l_2, l_3, ..., l_m)$, where $m < n$. In a typical MLaaS scenario, F is usually owned by the service provider and invisible (or unavailable) to customers. FD needs to utilize model F's middle layer's outputs when transforming noise into weak feature noise (WFN). In other words, it needs to abandon part of the original MLaaS service mode to achieve its purpose. As shown in Figure 1, F_{sub} is provided locally by service providers when customers purchase their services (different from the traditional way). Customers no longer directly upload their data to the cloud server when obtaining the service they purchased but provide WFN.

Our goal is to make the difference in statistical characteristics and visual representation between raw data and WFN as large as possible as follows.

$$\arg\max_{WFN} KL(P(raw), Q(WFN)) \tag{1}$$

where P and Q represent the distribution of raw data and noise WFN, respectively. In addition, the results returned by the cloud are also as similar as possible. Accordingly, we define the objective function as follows.

$$\arg\min_{WFN} F(raw) - F(WFN). \tag{2}$$

We also need to create an optimization method so that the above formula can be effectively solved. Feature dilution will be introduced in the next section.

B. Weak-feature Definition

To enable weak-feature noise (WFN) to have the ability to obtain services normally, we have proposed the following three binding characteristics, and any WFN should satisfy these characteristics. Also, intuitively, the idea of FD can be

applied to speech recognition, NLP, and other fields. Due to space limitations, we only verify it on images.

Characteristic 1: Label consistency. One of the most critical tasks in computer vision is classification, such as analyzing and judging patient medical images in intelligent medicine. Therefore, the cloud model's prediction result for the WFN must be the same as the original one. Otherwise, the returned result will be harmful to the customer. Nevertheless, the pixel changes in the image frequently change its label. Besides, the customer does not know the correct labels of the original samples during the desensitization process, which dramatically increases the difficulty of maintaining label consistency. Hence, in this paper, we list it as an essential property of WFN.

Characteristic 2: Feature consistency. The same type of data usually has multiple manifestations (i.e., different cars with various styles may all belong to trucks), but for a specific sample, its features are unique. We regard this relationship between samples and features as an injective relationship. Our goal is to ensure that the WFN and the original feature also have this injective relationship, reducing the feature aggregation of the same type of WFNs. This way can prevent the WFN from becoming the average of the global sample features (this will make the sample lose its uniqueness, which is likely to make the service unreliable). Specifically, we need to optimize the following functions.

$$\arg\min_{z} F_{sub,m}(x) - F_{sub,m}(WFN) \tag{3}$$

where x and WFN represent the original image and the weak feature noise, respectively, and F_{sub}, m denotes the sub-model obtained after the original model is truncated from the m-th layer.

Characteristic 3: Visual and statistical asymmetry. Much previous work on privacy leakage is to obtain privacy by restoring the visual representation of original data. In short, the reason for privacy leaks in MLaaS is that clients directly give visually distinguishable samples to dishonest third parties. Therefore, removing the sample's visual representation is also an essential step so that the WFN has visual asymmetry. In addition, to prevent the adversary from stealing privacy from the statistical level (or feature level), we also need to make the original data and WFN statistical features as different as possible.

Ideally, if WFN satisfies the above three properties, not only does it have no visual correlation with the original data, but it is still meaningless noise at the statistical level, and it can also enable customers to obtain effective services.

C. Feature Dilution

In this section, we present how FD makes the weak-feature noise meet these three conditions. In the process of ensuring the label consistency between the WFN and the original sample, customers do not have the real label about the latter, hence our idea is to maintain the relationship between them based on the features extracted by the cloud model. In particular, for a specific neural network model F, we can regard it as the following composite function.

$$F(x) = l_n\left(\ldots l_m\left(\ldots l_3\left(l_2\left(l_1(x)\right)\right)\ldots\right)\ldots\right), \tag{4}$$

where l_i denotes the various layers of the neural network, such as a convolutional layer in CNN. In a typical image classification system, data labels are frequently output by the last layer. However, before the data reaches the last layer, other layers are required to preprocess the data. In other words, if we can ensure that the intermediate results of a particular intermediate layer are consistent when processing the WFN and the original sample, theoretically, we can ensure that the final output result of the model is consistent.

Since the convolutional neural network is essentially a feature extractor, the output of the intermediate layer mentioned above can be regarded as the intermediate feature extracted by the model. Thus, in the process of approximating features, we make the WFN and the original sample have the similarity of internal features, which we call feature consistency. The optimization process is formulated as the following.

$$F(x) = F\left(FD\left(F_{sub,m}, z\right)\right) = F(WFN), \tag{5}$$

where FD(feature dilution) represents the image desensitization algorithm, which aims to transform Z into the WFN.

Since F_{sub} is an incomplete model, its output is the feature vector extracted from layer m. According to our previous analysis, to ensure the consistency of features, we can update the noise z through the output of F_{sub}. The specific goal is to optimize the following formula.

$$\begin{gathered} \arg\min_{z} c\_\text{loss} \\ c\_loss = F_{sub,m}(x) - F_{sub,m}\left(FD\left(F_{sub,m}, z\right)\right) \end{gathered} \tag{6}$$

In this way, the features extracted from the WFN in layer m are the same as the raw sample. Therefore, the consistency of sample labels and the uniqueness of features are guaranteed.

Now, we are able to get a primary WFN. However, we observed an interesting phenomenon: the WFN satisfying the consistency of labels and features are very similar to the original samples, most of them only show color differences and introduce some noises. Therefore, our next strategy is to maximize the visual difference between the original data and the WFN on the premise of ensuring consistency of label and feature. We will solve this problem from two perspectives.

To ensure no visual similarity between the two images, the easiest way to think of is to increase the difference between the corresponding pixels. Therefore, we consider using the mean square error loss as the objective function, as Equation 7.

$$sm\_\text{loss} = \arg\max_{z} \|x - z\|^2. \tag{7}$$

Then the semantics of an image is frequently formed by the regular arrangement and combination of pixels. Consequently, we utilize total variation loss to erase the semantic information.

Then there is the following loss Equation 8.

$$tv\_\text{loss} = \arg\max_{z} \sum_{k=0}^{c}\sum_{i=0}^{w}\sum_{j=0}^{h-1} z(i,j,k) - z(i,j+1,k) + \sum_{k=0}^{c}\sum_{j=0}^{h}\sum_{i=0}^{w-1} z(i,j,k) - z(i+1,j,k) \quad , \tag{8}$$

where z represents a picture with a dimension of $w \times h \times c$. The total variation loss can increases the difference between adjacent pixels, so that the entire image produces abundant dense noise, eliminating semantics.

By combining the different loss functions mentioned above, a final loss can be obtained. As follows.

$$\begin{aligned} &\arg\min_{z} \text{loss} \\ \text{loss} \ &= \alpha \times c\_\text{loss} \ - \beta \times sm\_\text{loss} - \chi \times tv\_\text{loss} \ . \end{aligned} \tag{9}$$

It is worth mentioning that during the experiment, we found that $\alpha : \chi = 100 : 1$ will have better results in most cases. Nevertheless, in a few exceptional cases, this ratio will cause the labels of some WFNs to lose consistency. At this time, we need to set the ratio to $1000 : 1$ and increase the β appropriately. Accordingly, we can decide which set of ratios to apply according to the image's complexity (such as contour information, color distribution, etc.). Hence the final TA algorithm as Algorithm. 1.

Algorithm 1 Transformation Algorithm

Input: $F_{sub,m}(\cdot)$: The truncated model;
z: Noise image;
x: Raw image
Output: weak-feature noise (WFN);

$target\_feature = F_{sub,m}(x)$
$image\_complexity = complexity(x)$
if $image\_complexity < threshold$ **then**
  $\alpha : \chi = 100 : 1$
else
  $\alpha : \chi = 1000 : 1$
$z = \sigma \times x + \gamma \times z$ s.t. $\sigma + \gamma = 1$
for i in $epoch$ **do**
  $intermediate\_feature = F_{sub,m}(z)$
  $c\_\text{ loss} = F_{\text{sub },m}(x) - F_{\text{sub },m}\left(TA\left(F_{\text{sub },m}, z\right)\right)$
  $sm\_\text{ loss} = \arg\max_z \|x - z\|^2$
  $tv\_loss = total\_variation(x, z)$
  $loss = \alpha \times c\_\text{loss} \ - \beta \times sm\_\text{loss} - \chi \times tv\_\text{loss}$
  $\Delta w = \frac{\Delta loss}{\Delta z}$
  $z = z - \mu \Delta w$
$WFN = z$
return WFN

We successfully obtained the ideal WFN through the above process. However, the fundamental of Algorithm 1 is to fit the features extracted by the intermediate layer l_m. In computer vision, it is manifested as the intermediate feature map fitting. The features of lm depend on the features extracted from the l_{m-1}. Therefore, we want to know whether the features will be leaked through the middle layer output.

In general, when fitting the features of the l_m, the Algorithm 1 tends to fit the features of the $l_1, l_2, ..., l_{m-1}$ at the same time. i.e., even if the visual representation is successfully erased, the internal features of WFN will still retain a large number of original features that can be extracted and recognized by different network layers. This poses a critical obstacle to privacy protection.

We concretize this question. In the process of generating WFN, the backpropagation (BP) [9] will find the shortest path to optimize (generate) it, and the shortest path is that retains a large number of internal features of original data. Accordingly, we need to introduce constraints in the optimization process to prevent the BP algorithm from being "lazy". We divide the feature fitting into two parts. One part is used to ensure the first two characteristics of the WFN, and the other part is used to erase excessive internal features. We only need to change the 10-th line of Algorithm 1 to the following.

$c\_\text{ loss}\_1 = F_{\text{sub },m}(x) - F_{\text{sub },m}\left(TA\left(F_{\text{sub },m}, z\right)\right)$
$c\_\text{ loss}\_2 = F_{\text{sub },n}(x) - F_{\text{sub },n}\left(TA\left(F_{\text{sub },n}, z\right)\right)$ s.t. $n < m$
$c\_\text{loss} \ = \alpha_1 \times c\_\text{loss} \ \_1 - \alpha_2 \times c\_\text{loss} \ \_2$

To ensure that no features other than the l_m are fitted, we set n to 1.

IV. Performance Evaluation

A. Validity Verification

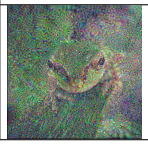
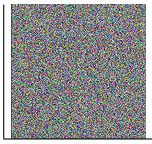
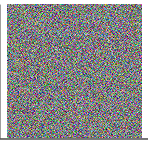
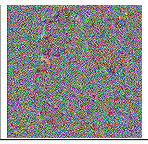
First, we need to test whether these three properties can guarantee the validity of the data. The experiment uses VGG19 that has been pre-trained on ImageNet as a service model. According to the experiment's needs, we changed the output of VGG19 to the intermediate layers' outputs and defined it as Fsub. We will apply the original VGG19 to classify the corresponding sample to determine whether the converted WFN is valid.

The desensitization process is shown in Table. I. It can be found that WFN, which only satisfies the third property, cannot provide effective services. Furthermore, the samples that meet the first and second conditions can replace the original samples to perform an effective query. However, they still leak the information of the original data because they contain too many visual representations.

B. Parametric Analysis

We randomly selected 1000 data and mixed the original sample with noise in different proportions (we call it the image-to-noise ratio) to study the initial sample's influence on the desensitization process. As shown in Table. II. We have found that in many cases, there is a trade-off between privacy and effectiveness. Experiments show that when $c\_loss\_weight : v\_loss\_weight = 1000$, the visual and statistical characteristics of most samples can be removed while still having high effectiveness. Also, when we conducted this

TABLE I: WFN generated under different constraints. The first row satisfies characteristics 1 and 2, the second row satisfies characteristic 3, and the third row satisfies all characteristics. F(raw) and F(WFN) represent the label obtained from VGG19, respectively.

Iters=500	Iters=1000	Ground Truth	F(RD)	F(MD)
			tree frog	tree frog
			quail	harp
			border collie	border collie

experiment, each sample only iterated 1500 times. However, we found that some samples require a longer iteration cycle, but generally no more than 4000 times (more than 4000 times, label reversal is very likely).

TABLE II: Effectiveness of samples under different parameters.

	$\sigma = 0.9, \gamma = 0.1$	$\sigma = 0.6, \gamma = 0.4$	$\sigma = 0.4, \gamma = 0.6$
$\alpha = 1e10$	94.63%	95.10%	96.32%
$\alpha = 1e9$	93.54%	93.34%	94.96%

We are also very interested in the change of the samples' labels during the desensitization process. We recorded the label change and its corresponding loss, as shown in Figure 2. By adjusting the image-to-noise ratio, we can observe the influence of different sample initial values on the conversion result. It is not tough to find that the larger the proportion of noise, the longer the conversion period, that is, the more difficult it is to ensure the consistency of sample labels. Nevertheless, the final WFN will meet the visual and statistical asymmetry more. It is worth noting that we did not use the label loss in the transformation process, it is only used for observation.

C. Security Analysis

Although the WFN is visually difficult to cause privacy leakage, will privacy be extracted by the network and cause privacy leakage?

The middle layer we used in the previous experiment was "block5 conv3" in VGG19. Next, we further experimented with other layers. As mentioned before, we will maximize the i-th layer features when fitting the features of the m-th layer, where $i < m$. In the experiment, we set i to 1. And observe the changes in the features extracted by "block3 conv2". The experimental results are shown in Table. III.

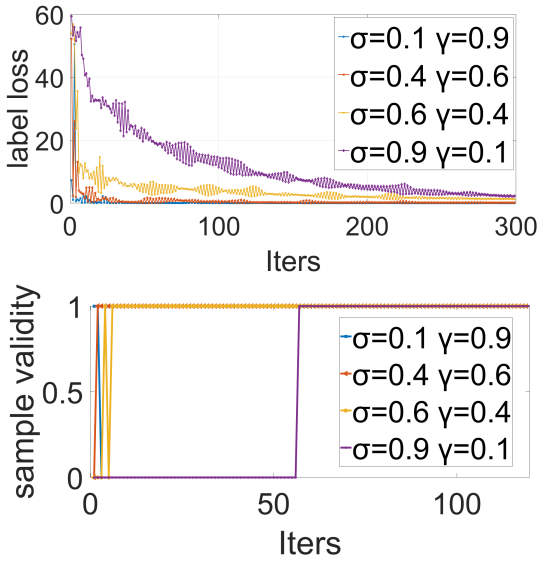
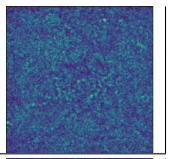
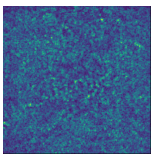
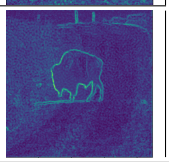
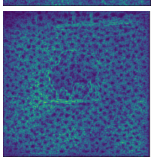


Fig. 2: The change of the WFN's label under different image-to-noise ratios during the transformation process.

TABLE III: Maximizing the difference in feature extraction and prevent privacy leakage at the feature level.

original feature	layer	raw features	processed feature
	block5_conv3		
	block3_conv2		

When we fit in-depth features, the features extracted in the first layer do not have any discernibility, thus privacy leakage cannot be caused at the feature level. When we leverage the shallow layer (e.g., block3_conv2) for fitting, the first layer's features will become very similar to the original features, which will cause the WFN to be quickly restored to the original sample. In this case, even if the WFN does not leak privacy, it can be stripped of its sensitive information characteristics by the neural network, resulting in privacy leakage. It is fascinating that the model does not know what the original data looks like during the feature extraction. Nevertheless, after maximizing the features extracted in the first layer, the leakage at the feature level is suppressed. Therefore, when choosing a specific l_m, the dimension should be as small as possible.

In addition to ensuring that the privacy of the client's local data is not leaked, we must also ensure that third parties cannot illegally use the WFN uploaded by clients. To this end, we randomly selected 1000 WFNs and tested their robustness. The

TABLE IV: Verifying whether WFN can be recognized by other networks and be used for secondary use.

Models	ACC on original data	ACC on WFN
VGG16	71.97%	19.43%
VGG19	72.69%	67.18%
resnet34	73.24%	15.84%
resnet50	76.26%	16.49%
resnet101	78.12%	12.61%
squeezenet1_0	56.95%	8.68%
densenet121	74.66%	15.57%
densenet169	75.35%	18.54%
densenet201	76.91%	18.11%
inception_v3	70.58%	13.63%

results are as Table. IV. The WFN can only be recognized by VGG19, which is consistent with the previous conjecture and experimental results. For other service models, WFNs are just a bunch of meaningless noise data. Therefore it is difficult for a third party to restore a small number of data features through these models. We think this can well protect client privacy.

Besides, we also consider whether a third party can use the WFN as a new training set to identify data uploaded by clients. We used VGG19 without pre-training for training, and the results are shown in the Figure. 3.

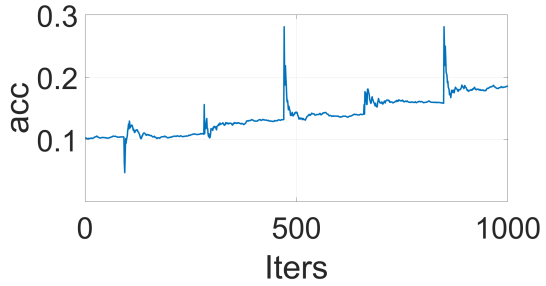


Fig. 3: The trend of training accuracy when using WFN to train VGG19.

We found that the model trained using WFN as the training set does not have any practical value. Even if VGG19 is used, the accuracy is less than 30%, and the training loss will eventually linger at 2. We speculate that the reason is that WFN is essentially a group of noise without any statistical rules, hence the neural network cannot find specific statistical rules from it, and ultimately the training accuracy cannot be improved. This experiment shows that WFN has high security once again.

V. Conclusions and Future Work

Although there are many methods to prevent privacy in machine learning as a service (MLaaS) scenarios, most methods utilize cryptography or a trusted platform. The former will cause much computational overhead, while the latter has insufficient privacy protection. This paper first introduced a service architecture that allows clients to obtain services without uploading private data. Then we designed an algorithm called feature dilution (FD) to help clients get services normally under this architecture. Experiments have shown that our proposed method is effective.

In the future, we intend to investigate how to protect the privacy of both clients and service providers more effectively. Meanwhile, we plan to further improve the service quality of the proposed scheme by learning more reasonable noise addition mechanisms.

References

[1] J Andrew Onesimu and J Karthikeyan. An efficient privacy-preserving deep learning scheme for medical image analysis. *Journal of Information Technology Management*, 12(Special Issue: The Importance of Human Computer Interaction: Challenges, Methods and Applications.):50–67, 2020.

[2] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.

[3] Jia Deng, Wei Dong, Richard Socher, Jia Li, and FeiFei Li. Imagenet: A large scale hierarchical image database. In *CVPR*, 2009.

[4] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[5] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N Wright. Privacy-preserving machine learning as a service. *Proceedings on Privacy Enhancing Technologies*, 2018(3):123–142, 2018.

[6] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. Chiron: Privacy-preserving machine learning as a service, 2018.

[7] Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. Secure outsourced matrix computation and application to neural networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1209–1222, 2018.

[8] Tong Li, Zhengan Huang, Ping Li, Zheli Liu, and Chunfu Jia. Outsourced privacy-preserving classification service over encrypted data. *Journal of Network and Computer Applications*, 106:100–110, 2018.

[9] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, pages 1–12, 2020.

[10] Xu Ma, Xiaofeng Chen, and Xiaoyu Zhang. Non-interactive privacy-preserving neural network prediction. *Information Sciences*, 481:507–519, 2019.

[11] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902. IEEE, 2015.

[12] 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.

[13] Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

[14] Mu Wang, Changqiao Xu, Xingyan Chen, Hao Hao, Lujie Zhong, and Shui Yu. Differential privacy oriented distributed online learning for mobile social video prefetching. *IEEE Transactions on Multimedia*, 21(3):636–651, 2019.

[15] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[16] Shui Yu. Big privacy: Challenges and opportunities of privacy study in the age of big data. *IEEE access*, 4:2751–2763, 2016.