DISTRIBUTED ONE-CLASS LEARNING

Ali Shahin Shamsabadi[⋆]*, Hamed Haddadi*[†]*, Andrea Cavallaro*[⋆]

[⋆]Queen Mary University of London,[†]Imperial College London

ABSTRACT

We propose a cloud-based filter trained to block third parties from uploading privacy-sensitive images of others to online social media. The proposed filter uses Distributed One-Class Learning, which decomposes the cloud-based filter into multiple one-class classifiers. Each one-class classifier captures the properties of a class of privacy-sensitive images with an autoencoder. The multi-class filter is then reconstructed by combining the parameters of the one-class autoencoders. The training takes place on edge devices (e.g. smartphones) and therefore users do not need to upload their private and/or sensitive images to the cloud. A major advantage of the proposed filter over existing distributed learning approaches is that users cannot access, even indirectly, the parameters of other users. Moreover, the filter can cope with the imbalanced and complex distribution of the image content and the independent probability of addition of new users. We evaluate the performance of the proposed distributed filter using the exemplar task of blocking a user from sharing privacy-sensitive images of other users. In particular, we validate the behavior of the proposed multi-class filter with non-privacy-sensitive images, the accuracy when the number of classes increases, and the robustness to attacks when an adversary user has access to privacy-sensitive images of other users.

***Index Terms*—** Distributed Learning, One-Class Autoencoder, Privacy

1. INTRODUCTION

Unauthorized sharing of potentially privacy-sensitive images of other users is an increasingly important privacy challenge in online social media. To protect privacy, a cloud sharing images should filter uploaded content to prevent the sharing of unauthorized (or undesired) privacy-sensitive images. However, to reach this goal a service provider would need to access the privacy-sensitive images themselves in order to produce a hash or to train a filter. This centralized learning solution only shifts and does not solve the problem of maintaining certain images private [1]. This privacy challenge could be addressed by distributed learning [2, 3], where each user (i.e. each edge device) uploads to the service provider only the *parameters* of their machine learning model (from a *local* copy of the service provider's learning model), not their raw images. The service provider then fine-tunes the *global* learning models by combining the values of the parameters of each user. Finally, the updated parameters are downloaded by the device of each user, which would consequently update their local learning models and upload the updated parameters back to the cloud. This iterative process stops when a certain classification accuracy is achieved.

While distributed learning can be considered more privacy-friendly than centralized learning (as a result of uploading parameters of local models instead of raw image dataset), it has important limitations. First, the parameters of each user are not only shared with the service provider, but also (indirectly) with the other users when the shared parameters are downloaded from the cloud. Through the analysis of the parameters during the training phase an adversary user can recover information about the training data of other users [4, 5]. Second, the user should have access to data of several classes as each user updates the parameters of a local copy of a multi-class classifier. This is in contrast to our scenario where each user only has access to data of their own class(es). Third, a new user cannot join the system during the test phase when other users are using the service, and therefore scalability is an issue.

To tackle these limitations, we propose *DOCL*, a Distributed One-Class Learning approach that decomposes the global filter down to N one-class classifiers distributed among the users. Each user trains an autoencoder as a classifier on their privacy-sensitive data. The autoencoder learns to reconstruct its privacy-sensitive training image with minimum error. The global filter then aggregates all the one-class autoencoders and, to discriminate between classes and block privacy-sensitive images, measures the dissimilarity between new images uploaded by (other) users and the reconstructions of the autoencoders.

The proposed filter has several desirable properties. The training of each one-class classifier is *independent* from that of the other classifiers and therefore training data and parameters are not shared among users, thus preserving the privacy of training data against adversary users. Moreover, as the global filter is decomposed into a series of simple one-class classifiers trained by the users themselves, even the service provider has no access to the privacy-sensitive images of the users. In addition to the above, the proposed filter can cope with *imbalanced training data*, which is an important property as the number of training images for the different classes is likely to be different. In particular, when the imbalance of the training data increases, the performance of one-class classifiers is almost stable, whereas that of, for example, multi-class classifiers (even binary classifiers) decreases [6]. Finally, the size of the global filter depends on the number of users, which can be easily increased by uploading their one-class classifiers trained on their privacy-sensitive images to the cloud at any point.

2. THE DISTRIBUTED LEARNING MODEL

2.1. The local classifier: training at the edge

One-class classifiers or data descriptors [7] learn to distinguish the target class from outlier classes (i.e. when only data of the target class are available) using density estimation [8], data reconstruction [9] or closed boundary estimation [10, 11, 12]. In our proposed framework, DOCL, each user trains as classifier a one-class autoencoder [9], a three-layer parametric neural network with an input layer, a data representation layer, and an output layer [13, 14, 15].

For simplicity but without loss of generality, let the number of classes correspond to the number of users, N. Let the users

$\{u_0, u_1, ..., u_{N-1}\}$ train independently N one-class autoencoders on their privacy-sensitive image datasets $\{I_0, I_1, ..., I_{N-1}\}$. Each u_i feeds their set $I_i = \{\mathbf{I}_{i,0}, \mathbf{I}_{i,1}, ..., \mathbf{I}_{i,j}, ..., \mathbf{I}_{i,K_i-1}\}$ to a ResNet [16], which is already trained by the service provider on public non-privacy-sensitive images (see section 3) in the cloud. Note that the cardinality of each set $|I_i| = K_i$ may considerably differ across users. For each I_i, ResNet generates the corresponding feature set, X_i, defined as:

$$X_i = \{\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, ..., \mathbf{x}_{i,j}, ..., \mathbf{x}_{i,K_i-1}\}, \tag{1}$$

where $\mathbf{x}_{i,j} \in \mathbb{R}^D$ and D is the feature dimension.

The parametric *encoder* of user u_i obtains $\mathbf{h}_{i,j} \in \mathbb{R}^M$, a feature representation whose dimension $M < D$, by applying a Rectifier Linear Unit $f(\cdot)$ [17] on the linear combination of the elements of feature $\mathbf{x}_{i,j}$:

$$\mathbf{h}_{i,j} = f(\mathbf{W}_i \mathbf{x}_{i,j} + \mathbf{b}_i) \quad \forall j = 0, 1, ..., K_i - 1, \tag{2}$$

where $\mathbf{W}_i \in \mathbb{R}^{M \times D}$ and $\mathbf{b}_i \in \mathbb{R}^M$ are the parameters of the encoder for user u_i. Then the parametric *decoder* of user u_i maps feature representation $\mathbf{h}_{i,j}$ to the feature reconstruction $\hat{\mathbf{x}}_{i,j} \in \mathbb{R}^D$ by applying a sigmoid function $g(\cdot)$ [18] on the linear combination of the elements of $\mathbf{h}_{i,j}$:

$$\hat{\mathbf{x}}_{i,j} = g(\mathbf{W}_i^{'} \mathbf{h}_{i,j} + \mathbf{b}_i^{'}) \quad \forall j = 0, 1, ..., K_i - 1, \tag{3}$$

where $\mathbf{W}_i^{'} \in \mathbb{R}^{D \times M}$ and $\mathbf{b}_i^{'} \in \mathbb{R}^D$ are the decoder's parameters for u_i.

Let $C(\cdot)$ represent the binary cross-entropy differences among features and feature reconstructions:

$$C(\mathbf{x}_{i,j}, \hat{\mathbf{x}}_{i,j}) = -\sum_{l=1}^{D} x_{i,j}(l) \log \hat{x}_{i,j}(l) + (1 - x_{i,j}) \log(1 - \hat{x}_{i,j}(l)), \tag{4}$$

where $x_{i,j}(l)$ and $\hat{x}_{i,j}(l)$ are the l-th elements of $\mathbf{x}_{i,j}$ and $\hat{\mathbf{x}}_{i,j}$, respectively. The final parameters for u_i, $\{\mathbf{W}_i^*, \mathbf{W}_i^{'*}, \mathbf{b}_i^*, \mathbf{b}_i^{'*}\}$, are learned with the following optimization:

$$\{\mathbf{W}_i^*, \mathbf{W}_i^{'*}, \mathbf{b}_i^*, \mathbf{b}_i^{'*}\} = \underset{\{\mathbf{W}_i, \mathbf{W}_i', \mathbf{b}_i, \mathbf{b}_i^{'}\}}{\arg\min} \sum_{j=0}^{K_i - 1} C(\mathbf{x}_{i,j}, \hat{\mathbf{x}}_{i,j}) \tag{5}$$

and are obtained using ADADELTA [19]. These parameters are then uploaded to the cloud for the global filter to populate the N-class classifier composed of N one-class[1] autoencoders. Note that unlike [2, 3], in our case the parameters of each user are *not* downloaded by other users.

Moreover, each user u_i uploads to the global filter also the parameters that describe the distribution of the differences, $d(\mathbf{x}_{i,j}, \hat{\mathbf{x}}_{i,j})$, between the input features and the reconstructed features for all training images, using the final parameters of their autoencoder. These differences are computed as:

$$d(\mathbf{x}_{i,j}, \hat{\mathbf{x}}_{i,j}) = \sum_{l=1}^{D} ||x_{i,j}(l) - \hat{x}_{i,j}(l)||_2^2, \quad \forall j = 0, 1, .., K_i - 1, \tag{6}$$

where $x_{i,j}(l)$ and $\hat{x}_{i,j}(l)$ are, respectively, the l-th elements of feature vector of a training image and that of its reconstruction. The edge device of each user, u_i, fits the distribution $d_{i,j} = d(\mathbf{x}_{i,j}, \hat{\mathbf{x}}_{i,j})$ $\forall j = 0, 1, .., K_i - 1$ with a normal distribution and uploads its mean, μ_i, and standard deviation, σ_i, to the global filter in the cloud.

[1]Note that this is the worst case of imbalanced training data, where data of each user belongs to one class only.

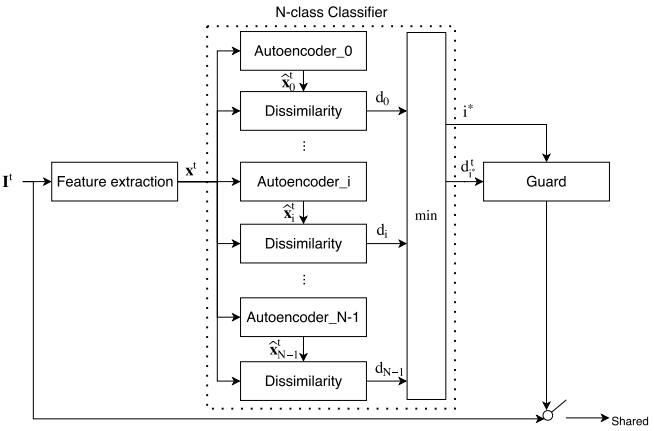

Fig. 1. Block diagram of the global filter. The service provider extract the features $\mathbf{x}^t$ from a newly uploaded image $\mathbf{I}^t$ passes them through the N one-class autoencoders, each trained independently by users. The minimum dissimilarity between $\mathbf{x}^t$ and its reconstructions $\hat{\mathbf{x}}^t$ is then analyzed by the Guard module prior to sharing $\mathbf{I}^t$.

2.2. The global filter

After aggregating the N locally trained one-class autoencoders in the global filter, users can use the service by uploading their images (see Fig. 1).

Let a user upload image $\mathbf{I}^t$ during the test phase. Prior to sharing that image, the service provider checks its legitimacy by feeding the features $\mathbf{x}^t$ generated from $\mathbf{I}^t$ to the global filter, which generates the reconstructed feature vector $\hat{\mathbf{x}}_i^t$ for each autoencoder:

$$\hat{\mathbf{x}}_i^t = g\Big(\mathbf{W}_i^{'*}\big(f(\mathbf{W}_i^* \mathbf{x}^t + \mathbf{b}_i^*)\big) + \mathbf{b}_i^{'*}\Big) \quad \forall i = 1, 2, .., N. \tag{7}$$

Each autoencoder reconstructs differently images of the same class as its training set and images of another class [9]. Hence the service provider quantifies the dissimilarity between the features of the uploaded image, $\mathbf{x}^t$, and of the reconstructed N feature vectors, $\hat{\mathbf{x}}_i^t$, generated by the N autoencoders, and determines the minimum:

$$d_{i*}^t = \min_{i=1,..,N} \sum_{l=1}^{D} ||x^t(l) - \hat{x}_i^t(l)||_2^2, \tag{8}$$

where $x^t(l)$ and $\hat{x}_i^t(l)$ are the l-th elements of feature vector of the uploaded image and of its i-th reconstruction, respectively.

The best reconstruction (i.e. the minimum dissimilarity score, d_{i*}^t) is expected for the class of the uploaded image belongs to. If the minimum dissimilarity does not correspond to the class of the user who has uploaded the image, then that image should be labeled either as privacy-sensitive image of another user (and therefore blocked) or as non-privacy-sensitive image for any of the contributing users (and therefore shared).

To decide whether an image should be blocked or shared, we propose a Guard module that the service provider uses to determine the *privacy interval* of each autoencoder using its μ_i and σ_i, which were estimated at the end of the training phase (see Sec. 2.1). The privacy interval of each user u_i is $[\mu_i - \alpha\sigma_i, \mu_i + \alpha\sigma_i]$, which defines the values of $d(\mathbf{x}^t, \hat{\mathbf{x}}_i^t)$ for which $\mathbf{I}^t$ should be considered a privacy-sensitive image of user i. The value of $\alpha \in [1, 2]$ defines the desired confidence value (see Sec. 3). If the minimum distance is outside this

	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
IMDB	402	414	347	528	559	198	414	704	619	554
CIFAR-10	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
MNIST	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949

Table 1. A sample image for each class and the number of training images from the IMDB, CIFAR-10, and MNIST datasets.

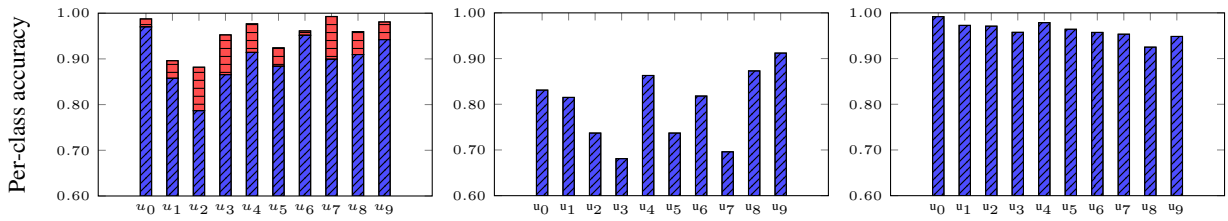

Fig. 2. Per-class accuracy of the global filter on IMDB (left), CIFAR-10 (middle) and MNIST (right) images. Note that for IMDB we used two autoencoders with representation layers of size 32 and 132.

interval, $\mathbf{I}^t$ is not considered as privacy-sensitive and hence shared. Otherwise, $\mathbf{I}^t$ is blocked by the Guard.

Note that each user can define multiple classes of privacy-sensitive images by training additional one-class autoencoders. Moreover, the proposed filter is scalable as each user individually trains a one-class classifier and this process takes place separately from that of other users. The number of users can, therefore, be increased in the test phase: new users train a one-class classifier on their privacy-sensitive images and upload the resulting parameters to the cloud to join the filter of the service provider. We will quantify the impact of increasing the number of users in the next section.

3. EVALUATION

To validate the performance of the proposed filter, we measure its overall accuracy, the per-class accuracy, the acceptance rate, the robustness to attacks and its scalability. As proof of concept that covers different classes a user might want to protect, we use three datasets (see Table 1): IMDB [20], CIFAR-10 [21] and MNIST [22]. We consider real limitations in these experiments such as different sizes of users' training dataset and limited training data. We assume $N = 10$ users, each represented by one class.

We randomly choose 10 celebrities from the **IMDB** dataset (5 different sets) as users/classes. We assume that the privacy-sensitive data are their faces and therefore we detect and crop the faces [23] with size $(3 \times 224 \times 224)$. The feature extractor (ResNet trained on non-privacy-sensitive ImageNet images) of each user (i.e. same for all users) u_i extracts a $D = 2,048$ dimension feature vector. We consider a smaller and a larger autoencoder which differ only in the size of their representation layer (32 and 132, respectively). Hence the size of input, representation and output layers are $2,048$, 32 (or 132) and $2,048$, respectively. To mimic the protection of classes of images other than people or faces, we use **CIFAR-10**, which contains 50K training and 10k test images $(3 \times 32 \times 32)$ of 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Users obtain a $D = 256$ dimension feature vector [24][2] from their images and hence the size of input, representation and output layers are 256, 32 and 256, respectively. Finally, we consider **MNIST** includes 60K training and 10K test $(1 \times 28 \times 28)$ images of 10 different handwritten digits. For example, u_0 has 5,923 images of digit 0 and u_1 has 6,742 images of digit 1 for training their one-class autoencoder. Due to the simplicity of this dataset, no feature extractor is used and the raw pixel values are given as input to the autoencoder. The size of input, representation and output layers of each user's one-class autoencoder are 784, 32 and 784, respectively.

We first quantify the *overall accuracy* as the ratio between the number of correctly predicted labels for all classes and the total number of test data of all of the classes; and the *per-class accuracy* as the ratio between the number of correctly predicted labels of each class and the total number of that class.

The overall accuracy of the global filter is 97% and 80% on MNIST and CIFAR-10, respectively; whereas the overall accuracy for IMDB with the smaller and larger autoencoder is 92% and 97%, respectively. The per-class accuracy is shown in Fig. 2. In MNIST the accuracy of digit 0 is over 99%, that of digit 9 is approximately 95%, as this digit shares similarity with digits 3, 4 and 7. The proposed filter performs better on MNIST and IMDB, which are more related to our application, than on CIFAR-10, which has a high inter-class variability.

Next, we quantify the *acceptance rate* of the Guard as ratio between the number of correctly shared non-privacy-sensitive images and the total number of uploaded non-privacy-sensitive images. We analyze the behavior of the global filter when a user, who contributed an autoencoder or is a new user, uploads non-privacy-sensitive images. We consider two scenarios: images that are substantially different from or are very similar to the privacy-sensitive images of registered users. The former scenario includes images of Fashion-MNIST [25], with $(1 \times 28 \times 28)$ fashion products' images of 10 categories (t-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag and ankle boot) in the MNIST experiment. The lat-

[2]Modified version of ResNet for data of the size $(3 \times 32 \times 32)$ trained on non-privacy-sensitive CIFAR-100 images.

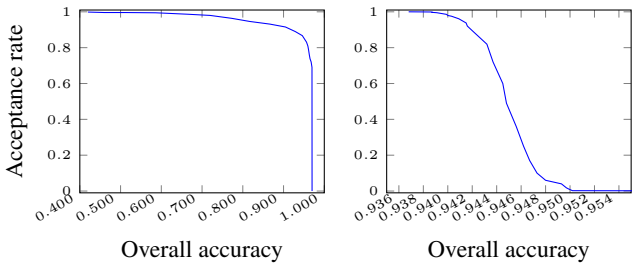

Fig. 3. Accuracy vs. acceptance rate on IMDB (left) and MNIST (right) when varying the confidence level $\alpha \in [1, 2]$ in the Guard.

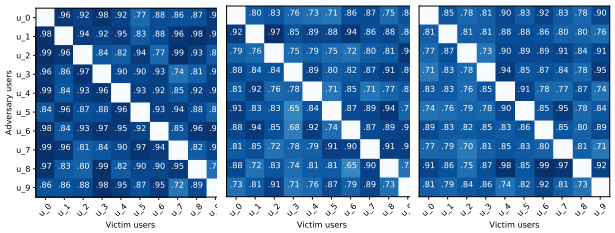

Fig. 4. Accuracy in detecting an adversary user in IMDB (left), CIFAR-10 (middle) and MNIST (right) with the global filter.

ter scenario includes faces with new identities (i.e. not used in the training phase) of IMDB for the IMDB experiment. Fig. 3 shows the relationship between sharing non-privacy-sensitive data (acceptance rate) and blocking privacy-sensitive images (overall accuracy). The larger the privacy interval in the Guard, the larger the overall accuracy and the lower the acceptance rate.

Moreover, we analyze the *robustness* of the global filter against attacks when an adversary gains access to the data of another user and attempts to share their privacy-sensitive images publicly. We assume that the adversary user has full knowledge (i.e. full access) of the training images of *some* users. This adversary user can train their local one-class autoencoder on the combination of his/her images with the available images of other users. Hence, the adversary user uploads the trained one-class classifier to the cloud in order to fool the service provider in sharing privacy-sensitive images of their targeted users. To quantify robustness we consider each user as an adversary and compute the accuracy of detecting and blocking that user (Figure 4). The rows and columns represent adversary and victim users, respectively. For example, when we consider u_3 as an adversary and u_2 as victim user, it means that u_3 trains his/her local one-class autoencoder with images of u_3 and of u_2. Then u_3 uploads the trained one-class autoencoder to the cloud. Sharing privacy-sensitive images of each user by other users is blocked in IMDB dataset better than with the other two datasets. This result stems from the dissimilarity of data from different classes. For example in MNIST, digits 2 shares considerable similarities with digit 7, so u_7 succeeds in nearly 3 out of 10 attempts of sharing images of u_2 when training a one-class autoencoder on data of digit 7 as well as digit 2.

Finally, to evaluate the *scalability* of the framework, we analyze the per-class accuracy trend when the number of classes increases. The per-class accuracy of the global filter for IMDB, CIFAR-10 and

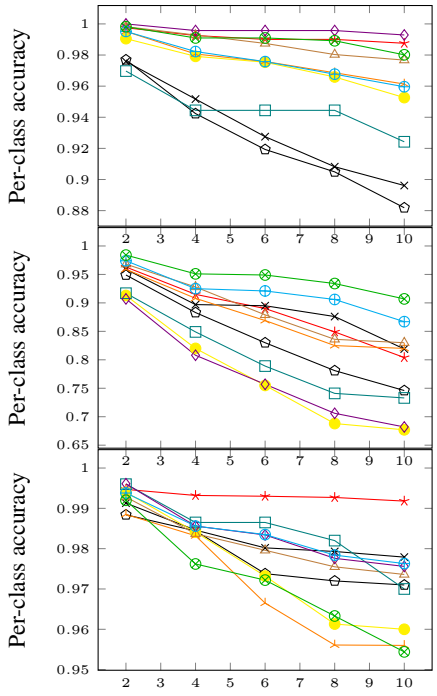

Fig. 5. The effect of increasing the number of classes in IMDB (top), CIFAR-10 (middle) and MNIST (bottom) on the per-class accuracy of the global filter for u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9.

MNIST for a varying number of classes is shown in Figure 5. The plots compare the accuracy of each class with different numbers of classes, i.e. different combinations of 1, 3, 5, 7, 9 classes. The influence on performance when increasing the number of classes on per-class accuracies in IMDB and MNIST is smaller than in CIFAR-10, because of its intra-class variability. The more similar the images in one class, the smaller the decrease in per-class and overall accuracy when the number of classes increases.

4. CONCLUSION

We presented a filter that aims to prevent a user from sharing to a social networking website privacy-sensitive images of other users without their consent. The proposed filter enables a centralized classification without the need of sending the training data to the central server/cloud as the training phase is performed independently for each user on their specific class(es).

This work is the first step in bringing one-class classifiers to distributed learning approaches in order to design a cloud-based filter with collaboration from the users with minimum computational costs and privacy loss. Giving sharing permission or blocking the uploading image is based on the comparison of the filter output and the user who has uploaded the image.

Future work includes extending the validation of the proposed filter on larger datasets and on different types of privacy-sensitive data beyond images.

Acknowledgment. This work was supported by a Microsoft Azure for Research grant (ref: CRM:0740917). Hamed Haddadi was also supported by the EPSRC Databox grant (Ref: EP/N028260/1).

5. REFERENCES

[1] O. Solon, "Facebook asks users for nude photos in project to combat 'revenge porn'." https://www.theguardian.com/technology/2017/nov/07/facebook-revenge-porn-nude-photos, 2017.

[2] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the Computer and Communications Security (CCS)*, pp. 1310–1321, ACM, 2015.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *in Proceedings of the International Conference on Articial Intelligence and Statistics (AISTATS)*, 2016.

[4] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," in *Proceedings of the Computer and Communications Security (CCS)*, pp. 603–618, ACM, 2017.

[5] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2017.

[6] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?," in *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, pp. 102–106, IEEE, 2012.

[7] D. M. J. Tax, *One-class classification*. PhD thesis, Delft University of Technology, 2001.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley, New York, 1973.

[9] L. Manevitz and M. Yousef, "One-class document classification via neural networks," *Neurocomputing*, vol. 70, no. 7, pp. 1466–1481, 2007.

[10] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[11] D. M. Tax and P. Juszczak, "Kernel whitening for one-class classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 03, pp. 333–347, 2003.

[12] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1096–1103, ACM, 2008.

[15] A. S. Shamsabadi, M. Babaie-Zadeh, S. Z. Seyyedsalehi, H. R. Rabiee, and C. Jutten, "A new algorithm for training sparse autoencoders," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 2141–2145, IEEE, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, 2016.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 315–323, 2011.

[18] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," *From Natural to Artificial Neural Computation*, pp. 195–201, 1995.

[19] M. D. A. Zeiler, "An adaptive learning rate method. arxiv preprint," *arXiv preprint arXiv:1212.5701*, 2012.

[20] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition.," in *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 1, p. 6, 2015.

[21] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*. Master's thesis, University of Toronto, 2009.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 529–534, IEEE, 2011.

[24] F. Chollet *et al.*, "Keras." https://github.com/keras-team/keras, 2015.

[25] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.