PICS: Private Image Classification with SVM

Eleftheria Makri
imec-COSIC, KU Leuven &
ABRR, Saxion University of Applied Sciences
emakri@esat.kuleuven.be

Dragos Rotaru
Bristol Cryptography group, University of Bristol
dragos.rotaru@bristol.ac.uk

Nigel P. Smart
Bristol Cryptography group, University of Bristol
nigel@cs.bris.ac.uk

Frederik Vercauteren
imec-COSIC, KU Leuven
frederik.vercauteren@esat.kuleuven.be

Abstract

*The advent of Machine Learning as a Service (MLaaS) makes it possible to outsource a visual object recognition task to an external (*e.g. *cloud) provider. However, outsourcing such an image classification task raises privacy concerns, both from the image provider's perspective, who wishes to keep their images confidential, and from the classification algorithm provider's perspective, who wishes to protect the intellectual property of their classifier. We propose PICS, a private image classification system, based on polynomial kernel support vector machine (SVM) learning. We selected SVM because it allows us to apply only low-degree functions for the classification on private data, which is the reason why our solution remains computationally efficient. Our solution is based on Secure Multiparty Computation (MPC), it does not leak any information about the images to be classified, nor about the classifier parameters, and it is provably secure. We demonstrate the practicality of our approach by conducting experiments on realistic datasets. We show that our approach achieves high accuracy, comparable to that achieved on non-privacy-protected data while the input-dependent phase is at least 100 times faster than the similar approach with Fully Homomorphic Encryption.*

1. Introduction

Visual object recognition is an important machine learning application, deployed in numerous real-life settings. Machine Learning as a Service (MLaaS) is becoming increasingly popular in the era of cloud computing, data mining, and knowledge extraction. Object recognition is such a machine learning task that can be provided as a cloud service. However, in most application scenarios, straightforward outsourcing of the object recognition task is not possible due to privacy concerns. Generally, the *image holder* who wishes to perform the image classification process, requires their input images to remain confidential (*i.e*. to not be revealed to the service provider). On the other hand, the *classification algorithm provider* wishes to commercially exploit their algorithm; hence, requires the algorithm parameters to remain confidential.

We consider an approach, which facilitates the outsourcing of the image classification task to an external classification algorithm provider, without requiring the establishment of trust, contractually or otherwise, between the involved parties. We focus on the evaluation task (*i.e*. labeling a new unclassified image), and not the learning task. Our proposal is based on Secure Multiparty Computation (MPC), and allows for private image classification without revealing anything about the private images of the image holder, nor about the parameters of the classification algorithm. Unlike previous work [2, 3, 16], we can fully outsource the task at hand, in such a way that the classification algorithm provider does not need to be the same entity as the *cloud computing provider*. This means that anyone with a valuable trained classifier can become a classification algorithm provider, without worrying about disclosing the parameters of their classifier.

PICS, our privacy-preserving image classification solution, combines the techniques of polynomial kernel support vector machine (SVM) classification, with the techniques of Secure Multiparty Computation (MPC). MPC allows a set

of mutually distrusting parties to jointly compute a function on their inputs, without revealing anything about these inputs (other than what can be inferred from the function output itself). MPC is based on a combination of cryptographic techniques, and secure communication amongst the computing parties, to achieve the aforementioned goal. Currently, MPC allows one to compute relatively simple functions on private data. Arbitrarily complex functions can be supported, but with large computational cost. This is why we focus on classification via SVM, as opposed to using more sophisticated techniques, such as Neural Networks.

Specifically, we have implemented our solution using SPDZ [5], which was introduced by Damgård *et al.* [12, 11], and is based on additive secret sharing. More details on the MPC techniques follow in Section 1.1. Unlike differential privacy techniques, which add noise to the inputs to preserve privacy, our solution enjoys provable security guarantees. A schematic representation of the application scenario treated by PICS is given in Figure 1. Using additive secret sharing techniques both the *classification algorithm provider*, and the *image holder* share their inputs to the $n \geq 2$ *MPC servers*. Note that no information about the actual secret inputs can be gained by the individual shares alone. Thus, each MPC server learns nothing about the inputs of the two parties. The cluster of the MPC servers comprise the *cloud computing provider*, which together execute the MPC protocol to produce the final classification result. The MPC servers communicate via authenticated communication channels to accomplish what the protocol prescribes. Finally, the protocol completes its execution by having all MPC servers sending their share of the final classification result to the party prescribed by the protocol, who can then reconstruct the result by combining the received shares. This party can be the *image holder*, or an external *analyst*, assigned to examine the classification results, without getting access to the underlying private images. Note that the involved parties, namely the image holder, the classification algorithm provider, and potentially the analyst, may play the role of the MPC servers themselves, avoiding completely the outsourcing to the cloud provider(s).

Our proposal is a first step towards implementing practical privacy-preserving image classification. We consider support vector machines trained with polynomial kernels, as they have been shown to outperform the accuracy of linear SVM [2]. Although Convolutional Neural Networks (CNNs) are at present the most prevalent machine learning method for image classification [19], SVM are favored over neural networks (NNs). This is because transforming a NN to a privacy-preserving one would result in an inefficient solution (*e.g.* 570 seconds for one image classification by CryptoNets [16]), given the non-linear nature of NNs. Mohassel, and Zhang [30], implement NN training, and evaluation, but only in the restricted setting of two-party computation, and for the weak security notion of passive security. We stress that our protocol works for an arbitrary number of MPC parties, and in the more stringent active security setting. As shown in Table 1, we are the only work providing active security. Additionally, our experiments demonstrated that PICS is computationally faster than any previous work.

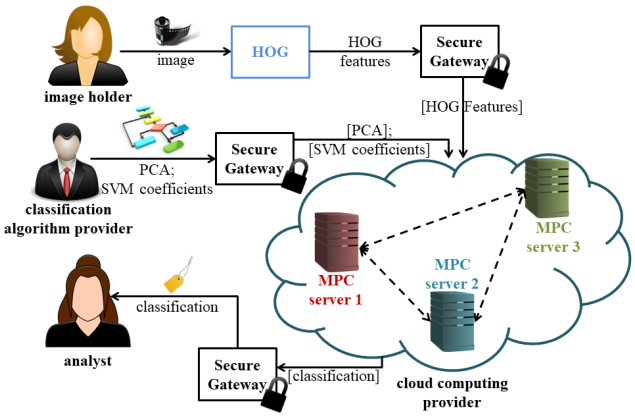

Figure 1. A schematic representation of the private image classification scenario.

Contributions:

- We enable full outsourcing of privacy-preserving image classification to a third independent party.
- Our solution does not leak *any* information about the private images, nor the classifier, while being the first to provide active security.
- We demonstrate the practicality of our approach, both in terms of efficiency, and in terms of accuracy, by conducting experiments on realistic datasets.

1.1. Preliminaries on Secure Multiparty Computation

Secure Multiparty Computation (MPC) is a cryptographic method allowing a set of parties to jointly compute a function on their inputs, without revealing the inputs to the rest of the parties. The two main security models used to realize MPC are the *passive*, and *active* security model. In the passive security model, we assume that the protocol participants follow the protocol specification honestly, but they try to learn as much information as possible about the private inputs, during the protocol execution. Under this model, the protocol participants are called honest-but-curious. In the active security model, we assume that the protocol participants may actively, and arbitrarily deviate from the protocol specification. Under this model, the protocol participants are called malicious. Clearly, the active

security model offers stronger security guarantees. In both models we can construct protocols that require an honest majority of the protocol participants to guarantee security, or protocols that guarantee security assuming a dishonest majority of the protocol participants. Our solution offers strong security guarantees, providing active static security, with a dishonest majority. This means that an adversary may corrupt, prior to the protocol execution, up to $n-1$ out of the n protocol participants, without leaking any private information, and without allowing any false protocol output to be accepted as correct.

Our solution is implemented using the SPDZ [12, 11] MPC framework, and that is why it enjoys the aforementioned security properties. The computational and communication costs of the constructed protocols increase linearly in the number of protocol participants. For our experiments we assume the minimum number of MPC servers necessary to perform the outsourced computation, namely two MPC servers, while scaling to more than two servers is straightforward. We consider this to be a reasonable assumption, given that these servers can be provided by independent cloud providers, such as Google, and Amazon (and Azure if we wish to expand to three parties), who have no incentive to collude against their clients. The probability that all cloud providers get corrupted by an adversary simultaneously is small.

SPDZ is based on additive secret sharing, allowing the participants to share their private inputs, in such a way that no information about the private inputs is revealed to the individual participants. Additive secret sharing enjoys an additively homomorphic property, meaning that any linear function can be directly computed on the shares that each protocol participant holds, without requiring interaction amongst the parties. Upon reconstruction of the shared output (which requires all parties to send their shares of the secret), the result will be the correct result of the linear function, as if it had been applied on the secret input. Thus, we can perform additions, and multiplications with non-secret constant values on the secret shared inputs. To perform multiplications between secret shared inputs, or any other non-linear operation, we need to execute a secure interactive protocol between the MPC servers.

The SPDZ approach works in two phases: a preprocessing phase, and an online phase. The preprocessing phase can take place offline, anytime prior to the execution of the online phase. This phase only requires the MPC servers to be online, and not the inputting parties. During this phase the MPC servers create shared randomness, which the client, and the classification algorithm provider can later use to securely share their private inputs. Moreover, the MPC servers create shared random values to be consumed during the online phase, and make it efficient. For the online phase, the two inputting parties first need to provide the MPC servers with their private inputs. This is performed in a secure manner, based on the Output Delivery protocol, and Input Supply protocol, proposed by Damgård *et al*. [10]. Then, the MPC servers proceed with the secure computation of the actual function, as prescribed by the protocol transcript. For more details on the MPC techniques used, we refer the reader to the work of Damgård *et al*. [12, 11].

2. Related Work

The related work on privacy-preserving machine learning focuses on providing a secure training phase, a secure classification phase, or both a secure training and classification. The majority of the research aims at designing a privacy-preserving training phase. Recently, due to the advent of cloud computing, and Machine Learning as a Service, more and more works focus on the design of a privacy-preserving classification phase. Fewer works have attempted to address both the training, and the classification phases in a privacy-preserving manner. Given the sheer number of works on private machine-learning, we only consider the works most related to ours in this section.

To facilitate an easy comparison of the related work, we summarize the main features of each proposal in Table 1. The first column of Table 1 is the reference to the corresponding research paper. The second column indicates whether the work considers a secure training phase (T), a secure training and classification phase (T+C), or only a secure classification phase (C). The third column indicates the security model, under which the proposed protocols are secure, where P stands for passive security, A stands for active security, and N/A (not applicable) refers to differential privacy techniques, which do not provide provable security guarantees. The fourth column denotes the method used to preserve privacy, where DP stands for differential privacy techniques; SP stands for selective privacy, and refers to the unique characteristic of the work of Shokri and Shmatikov [36] allowing the users to decide how much private information about their learned models they wish to reveal; MPC stands for Multiparty Computation; SHE stands for Somewhat Homomorphic Encryption; and 2-PC stands for Two-Party Computation. The fifth column lists the training method(s) used, where N-L SVM stands for non-linear Support Vector Machine; NN for Neural Networks; LM for Linear Means classification; FLD for Fisher's Linear Discriminant classification; HD for hyperplane decision; LIR for linear regression, LOR for logistic regression, and DT stands for decision trees. The sixth column lists the information that is revealed by the protocol execution. C stands for information about the classifier; TD stands for information about the training data; and CL stands for classification label. We note with boldface letters the information that is intentionally revealed by the protocol execution, and we mark with an asterisk the information that can poten-

tially, and unintentionally be leaked by the protocol execution. The last column indicates whether an implementation and experimental results of the suggested method have been provided.

	Func.	Model	Privacy Mthd	Train Mthd	Info Leak	Impl.
[25]	T	N/A	DP	N-L SVM	**C**; TD*	✓
[26]	T	N/A	DP	N-L SVM	**C*** TD*	✓
[36]	T	P	SP	NN	**C**	✓
[38]	T	P	MPC	N-L SVM	**C**	✓
[37]	T	P	MPC DP	N-L SVM	**C**	✓
[7]	T	P	MPC DP	NN	no*	✓
[18]	T+C	P	SHE	LM; FLD	no*	✓
[1]	T+C	P	SHE	Bayes; random forests	no	✓
[23]	T+C	P	2-PC	N-L SVM	**C** CL*	×
[8]	T+C	P	2-PC	NN	TD*	×
[30]	T+C	P	2-PC	NN LIR LOR	no	✓
[16]	C	P	SHE	NN	no	✓
[2]	C	P	SHE	N-L SVM	**C***	✓
[4]	C	P	SHE 2-PC	HD; Bayes; DT	no	✓
[33]	C	P	2-PC	N-L SVM	no	✓
[3]	C	P	2-PC	NN	**C**	×
[31]	C	P	2-PC	NN	no*	✓
PICS	C	A	MPC	N-L SVM	no	✓

Table 1. Comparison of the Related Work

Training a SVM in a privacy-friendly way, has been considered based on techniques of differential privacy [25, 26]. Despite the little overhead that these techniques incur, which makes them competitive from an efficiency perspective, they do not provide provable security guarantees. Shokri and Shmatikov [36] achieve privacy-preserving collaborative deep learning with multiple participants, while refraining from using cryptographic techniques. Although their work focuses on learning the artificial neural network, they do consider protecting the privacy of each individual's neural network, and allow the participants to decide how much information they wish to share about their models.

A lot of research has been devoted to *provable* privacy-preserving techniques for training a classifier. This line of research, much like ours, originates from Yao's millionaire problem [40], describing two-party computation, and its extension to multiparty computation [17] to securely compute any generic function. Specifically, the challenge of privacy-preserving data mining has been an active research area since the seminal work of Lindell and Pinkas [27]. More recently, Vaidya *et al*. [38] showed how to train a SVM classifier, in a privacy-preserving manner, based on vertically, horizontally, and arbitrarily partitioned training data. In follow-up work, Teo *et al*. [37] improved upon the efficiency of the solution of Vaidya *et al*. [38], and showed that their approach scales well to address the challenges of data mining on big data. Chase *et al*. [7] combine MPC techniques with differential privacy techniques to achieve private neural network learning. Their work provides provable security guarantees for the learning phase (though in the passive security model), and adds noise to the final resulting network to protect its privacy.

A parallel research line aiming to address the same challenge, namely privacy-preserving data mining, is based on homomorphic encryption (instead of MPC). The notion of homomorphic encryption dates back to the work of Rivest *et al*. [34], but only recently fully homomorphic encryption was devised [15]. This type of homomorphic encryption allows the computation of any polynomial function on the encrypted data, and unlike MPC, it does not require communication, as the task can be outsourced to one single party. Since the seminal work of Gentry [15], a lot of somewhat homomorphic encryption schemes have been proposed, allowing computations of polynomial functions of a limited degree. Graepel *et al*. [18] consider both machine learning training, and classification based on encrypted data, with their solutions being secure in the passive model. Due to the selected homomorphic encryption scheme, Graepel *et al*. [18] cannot treat comparisons efficiently, which excludes SVM-based solutions. Addressing both learning, and classification based on extremely random forests, and naïve Bayes networks, Aslett *et al*. [1], also work on homomorphically encrypted data.

One of the first private SVM classifiers was proposed by Laur *et al*. [23], which addresses both the training and the classification in a privacy-preserving manner. Their work combines the techniques of homomorphic encryption, secret sharing, and circuit evaluation, into a passively secure 2-PC solution. Concurrently, and independently Dahl [8] is working on using the very same MPC framework as in our work, to realize both the training, and the classification phase of CNN based privacy-preserving algorithms. While Dahl [8] is deploying CNNs instead of SVM, he needs to apply them in a non-black-box fashion. The protocol of Dahl [8] allows some leakage of information during the

training phase, which is not the case with our approach. Mohassel, and Zhang [30] also consider both training and classification in the 2-PC setting, and the passive security model. These approaches [23, 8, 30] can only treat the two-party setting, and cannot be trivially extended to allow the classifier provider to be a different entity than the cloud provider.

Fewer works focus particularly on the private image classification problem, instead of the training of the model. Gilad-Bachrach *et al*. [16] propose a solution applicable to the image classification problem, based on homomorphically encrypted data. The resulting *CryptoNets* [16] provide an accuracy of 99%, and can make on average 51739 predictions per hour. However, this is only the case when the predictions are to be made simultaneously; for a single prediction the task takes 570 seconds to complete. Recent work [2] demonstrated the potential of polynomial-kernel SVM to be used for classification in a privacy-preserving manner. Specifically, Barnett *et al*. [2], apply the same machine learning techniques as in our work, but on encrypted data. Although, Barnett *et al*. [2] mention the potential of an MPC approach to be more efficient in such a setting, they do not consider it, because direct translation of the protocols to MPC would require interaction between the client and the classification algorithm provider during the computations. We overcome this limitation by extending the application scenario in such a way that allows the classification task to be fully outsourced to a cluster of independent third parties. In addition, our solution, unlike the one of Barnett *et al*. [2], protects the classifier parameters during the feature extraction phase, and performs a secure comparison (instead of revealing a masked value to the client). This way our solution is guaranteed to not leak any information to the client about the classifier parameters, while it is also orders of magnitude faster.

In the 2-PC setting, Bost *et al*. [4], and Rahulamathavan *et al*. [33] focus on the problem of private classification, where both the classifier parameters, and the client's input to be classified need to remain private. Both approaches offer passive security, and do not consider nor experiment with polynomial kernel SVM. Barni *et al*. [3] propose private Neural-Network (NN) based data classification, also in the 2-PC setting and passive security model. They suggest three protocols, which offer different privacy guarantees for the classifier owner, while always protecting fully the client's input. Follow up work by Orlandi *et al*. [31], also considers NN-based data classification extending the work of Barni *et al*. [3] in terms of privacy.To the best of our knowledge, we are the first to provide a privacy-preserving image classification tool based on polynomial-kernel SVM, offering active security. PICS is more efficient than previous work and achieves high accuracy comparable to the accuracy achieved on the cleartext data. More interestingly, PICS is not limited to the 2-PC setting, allowing a broad range of application scenarios to be treated by our solution.

3. PICS

The proposed private image classification solution, PICS, employs Histogram of Oriented Gradients (HOG) for the feature extraction and combines it with Principal Component Analysis (PCA) dimensionality reduction. Note that already the dimensionality reduction takes place in a privacy-preserving manner, protecting the inputs of both the client, and the classification algorithm provider. This prepares the inputs for the next stage, which is the deployment of the Polynomial Kernel Support Vector Machine (SVM). Our method assumes that the training of the classifier is performed on cleartext data, prior to our protocol execution.

Specifically, the classification process, depicted in Figure 1, starts with the image holder, who has an image to be classified. The first step consists of the HOG feature extraction and is performed by the image holder locally on the cleartext version of the image. The resulting HOG features are then shared (via the secure gateway) to the MPC servers by the image holder and thus are kept secret from the classification algorithm provider. We indicate secret shared (and thus protected) data in square brackets (Figure 1). The classification algorithm provider has already trained their polynomial kernel SVM classifier. The necessary parameters for the PCA dimensionality reduction and the SVM classification, which are the two subsequent steps, are shared to the MPC servers by the classification algorithm provider and are never revealed to the image holder (nor the analyst). In the following we detail the aforementioned steps while emphasizing which operations are performed securely.

3.1. HOG Feature Extraction

The first step that needs to be taken by an image holder is HOG feature extraction. HOG is a popular feature extraction method, proposed by Dalal and Triggs [9] and used by numerous SVM applications ever since (see for example [28, 14, 32]). For this feature extraction, the image is divided in non-overlapping regions, called cells, and for every pixel within each cell a histogram of gradients is created. The gradient of each pixel expresses a magnitude and a direction (of the intensity), and then the histogram is grouped into bins. The number of bins times the number of cells per image defines the number of HOG features. HOG feature extraction has to be performed locally by the image holder on the cleartext data. Since this step does not require any classifier parameters, there are no privacy concerns raised by its execution.

3.2. PCA Dimensionality Reduction

The main criticism against HOG descriptors is the relatively high computational load that they incur. To allevi-

ate this computational load, HOG feature extraction is often combined with dimensionality reduction techniques such as the Principal Component Analysis (PCA) [35, 20, 29] that we also deploy. Dimensionality reduction aims at reducing the computational cost of working with high dimensional data. It does so by projecting the data to a lower dimensional space, while retaining as much as possible linearly uncorrelated features, namely the principal components. The PCA dimensionality reduction is calculated as follows:

$$\mathbf{z} = A^{\intercal} \cdot (\mathbf{h} - \mathbf{m_h}), \quad (1)$$

where A is the covariance matrix of the training data, $\mathbf{m_h}$ is the vector of means of the training data, and $\mathbf{h}$ is the vector of HOG features.

To perform PCA we need both the classifier parameters A, $\mathbf{m_h}$, and the features extracted from the image holder's images. This step has to be performed in a privacy-preserving manner, to protect both the classifier's intellect and the privacy of the client's images. First, the classification algorithm provider secret shares A, and $\mathbf{m_h}$ to the MPC servers and then the image holder secret shares $\mathbf{h}$ to the MPC servers. Having received all the aforementioned shares, the MPC servers engage in an interactive protocol (the SPDZ online phase) and securely calculate $\mathbf{z}$.

3.3. Polynomial Kernel SVM Classification

SVM classification is one of the most popular classification methods in computer vision. Despite the increasing popularity and high effectiveness of CNN classification techniques, their deployment requires large training datasets [13] that are potentially difficult to obtain when the underlying data is privacy sensitive. In addition, black-box transformation of these methods to their privacy-preserving equivalents will result in classifiers that are computationally prohibitive to use. Thus, we opted for the design of a private SVM classifier. Specifically, we deploy classification based on a model trained with a polynomial kernel SVM, as these have been shown to outperform linear SVM [2].

To classify a new unlabeled input with our classifier trained with polynomial kernel SVM, we need to securely evaluate the sign of the following equation:

$$\mathsf{class}(\mathbf{z}) = \mathsf{sign}[b + \sum_{i=1}^{n} a_i \cdot y_i \cdot (1 + \mathbf{x_i} \cdot \mathbf{z})^d], \quad (2)$$

where:

- $\mathbf{z}$ is the vector calculated securely in the PCA dimensionality reduction step and is available at the MPC servers in shared form;
- b is the model intercept (also known as bias), calculated by the classification algorithm provider during the learning phase and secret shared to the MPC servers;
- a_i are the solutions (Lagrange multipliers) of the optimization problem addressed by the SVM;
- y_i are the class labels of the support vectors;
- $\mathbf{x_i}$ are the n support vectors; and
- d is the order of the polynomial kernel used.

The only non-secret information in the above list is the order of the polynomial kernel d. We could keep the d value secret as well, but at a large increase in the computational cost. The additional security gain from not revealing the degree of the non-linear SVM is not considered to be worth the large computational cost. The a_i and y_i are amongst the most valuable classifier parameters that the classification algorithm provider wishes to protect. To avoid the additional multiplications and thus increase performance, we expect the classification algorithm provider to first calculate $(a_i \cdot y_i)$, and then secret share these resulting values to the MPC servers. Having all the necessary values available, the MPC servers securely calculate the classification label for a new input, following Equation 2. Note that unlike previous work [2, 16], we do not treat the classifier parameters in the clear, but in a secret shared form.

4. Experiments

We evaluate PICS both in terms of accuracy and in terms of performance (*i.e.* execution time), on two typical benchmark datasets. Note that the classification accuracy achieved by PICS is identical to the accuracy of the same SVM classifier applied on cleartext data, up to three decimal places. Given that we only report the accuracy with a precision of up to two decimal places, we do not need to discern between the accuracy of PICS and the accuracy on cleartext data.

4.1. Experimental Setup

The first parameter to be determined is the number of MPC servers that we have at our disposal. Our experiments are conducted on two MPC servers, which yields the most efficient solution, but the proposed system can be trivially modified to be executed on more than two MPC servers. We assume a protocol-independent, input-independent preprocessing phase that takes place prior to the actual protocol execution between the MPC servers. The inputting parties do not need to be aware, nor contribute to this phase. This offline preprocessing phase creates the necessary randomness to boost the efficiency of the online phase of the protocol, and allows the inputting parties (image holder and classification algorithm provider) to securely share their inputs.

The online phase begins with the image holder and the classification algorithm provider sharing their inputs (HOG features, and PCA + SVM parameters, respectively) to the

MPC servers. This is performed by executing an interactive protocol between each inputting party and the two MPC servers, as Damgård *et al*. [10] proposed. Then, the actual private image classification task is executed between the two MPC servers only. It starts with the PCA dimensionality reduction, as described in Section 3.2, and continues with the classification of a new image input, as described in Section 3.3. In the end, each MPC server sends their resulting share to the image holder, or the analyst, who can then combine the shares and reconstruct the cleartext result, which is the desired class label.

We executed our experiments, simulating the two MPC servers on two identical desktop computers equipped with Intel i7-4790 processor, at 3.60 GHz over a 1Gbps LAN with an average round-trip ping of 0.3ms. Note that previous works [2, 16] did not take the communication cost (incurred by sending the client's inputs to the classification algorithm provider) into account in their performance analyses whereas for us it is accounted automatically by timing the pre-processing and the online phase.

4.2. MPC Cost Affecting Parameters

Given that the underlying MPC platform that we use to implement PICS is based on additive secret sharing, we can compute linear functions on the shared data almost for free. What significantly increases the computational cost of the privacy-preserving, MPC version of the image classification is the number of multiplications that need to be performed. Recall also that multiplications require online communication to be executed.

To calculate the PCA dimensionality reduction, as in Equation 1, given that the matrix A and the vectors $\mathbf{h}$ and $\mathbf{m_h}$ are in a shared form, we need to perform ($\#\mathsf{PCA\_features} \times \#\mathsf{HOG\_features}$) secure multiplications. To securely compute the final classification label in Equation 2, we need to perform $[n \times (\#\mathsf{PCA\_features} + \#\mathsf{mults\_for\_degree} + 1)]$ multiplications in the secret shared domain (plus an additional constant number of multiplications for the computation of the sign function). Recall that we possess the value d in cleartext form and note that the required number of multiplications to compute the power d, is not simply $d-1$. This is because we can employ addition-chain exponentiation to minimize the required number of secure multiplications. Note also that $a_i \times y_i$ has been precomputed before being secret shared by the classification algorithm provider, saving in total n multiplications.

4.3. Datasets

We selected two of the most popular datasets of natural images, namely CIFAR-10 [22] and MNIST [24], to perform our experiments on, so that our accuracy and timing results are representative. CIFAR-10 [22] is a dataset of 60000 32×32 color images, out of which 50000 are training images and 10000 are test images. CIFAR-10 features 10 classes of objects, with 6000 images per class. MNIST [24] also is a dataset of 60000 handwritten digit images, consisting of 50000 training images and 10000 test images. The images are grey-scale, they have been normalized to fit a 20×20 pixel box, and are centered in a 28×28 image.

4.4. Training

We performed the training on the cleartext datasets described in Section 4.3. We trained a binary SVM classifier, distinguishing between airplanes and automobiles on the CIFAR-10 dataset, and a binary classifier distinguishing between zeros and ones on the MNIST dataset. For both binary SVM classifiers we have used 5000 training samples per class and 1000 test samples per class afterwards for the classification accuracy assessment.

In addition, we trained multiclass SVM classifiers for all ten classes in the CIFAR-10 and MNIST datasets, based on the one-versus-all strategy (OvA) [39]. We experimented separately with classifying instances into any number of classes between three and ten, following the OvA technique. For training the multiclass SVM, we need to actually train as many individual SVM classifiers, as the number of classes C that we wish to take into account. To avoid overfitting, for each individual SVM we considered the total number of samples in the training dataset for the class in question (*i.e*. 5000 samples), and as many random samples from the remaining $C-1$ classes, equally divided amongst the classes (*i.e*. $[5000/(C-1)]$ samples per class). For each dataset and number of classes C denote this procedure as CIFAR-C and MNIST-C.

4.5. Classification Accuracy and Efficiency Results

We evaluated the computational performance, and classification accuracy of PICS on several combinations of the algorithm parameters, namely the number of classes to distinguish, the number of support vectors, the number of PCA features, and the order of the classifier. Note that Table 2 reports the parameters which maximize the accuracy upto 10^{-2} for each SVM multiclass experiment: MNIST-2 to MNIST-10. The ties are resolved by choosing the one with lower cost to evaluate in MPC. In our experiments the polynomial degree d of the SVM ranges from 1 to 7 and the number of PCA features from 5 to 50.

In the same manner we obtain online timings for the CIFAR-2 to CIFAR-10 which range from 0.3s (90% accuracy) to 6.7s (58%) accuracy with a similar precision-recall score to the accuracy. This is due to the fact that CIFAR is a tougher dataset to learn for an SVM.

The combination of highest accuracy and highest computational performance was achieved by the binary PICS classifier on the MNIST-2 dataset, achieving 99.9% accuracy, and 30ms online execution time. An optimization

which played an important role was to multi-thread wherever it was possible, in this way improving the run-times have improved by at least 6 times from the single threaded version. In Figure 3 we selectively report the accuracy for the easiest case of a binary classifier, the average case of a 5-classes classifier and the hardest case of a 10-classes classifier. For completeness in Figure 2 we estimate the input independent phase using triples and bits generated with MASCOT [21] in $\mathbb{F}_p$ where p is a 128 bit prime.

It can be easily seen from Figure 3 that increasing only in one dimension (PCA features or support vectors) can actually decrease the accuracy of the SVM. To improve accuracy one has to increase both dimensions until the PCA features become redundant. Another interesting fact that comes from the graphs is that PICS performance is well correlated with the SVM accuracy. It seems that one can only have good accuracy at the expense of increasing PICS run-times.

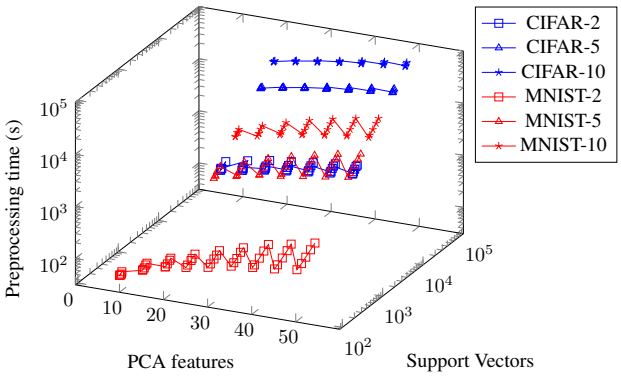

Figure 2. Offline cost for binary, 5-class, and 10-class classification.

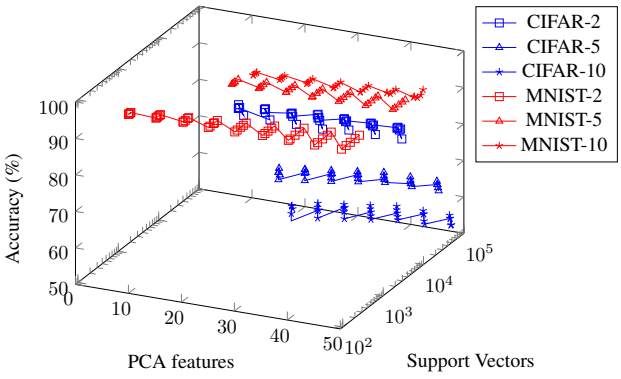

Figure 3. Accuracy of binary, 5-class, and 10-class classification.

5. Conclusion and Future Work

We have introduced PICS, a private image classification system, based on polynomial-kernel support vector machines. We showed how to achieve privacy-preserving image classification in such a way that the task can be fully outsourced to a third, independent party. For our solution we deployed generic MPC tools and showed how to avoid the restricted two-party setting. Unlike all previous work, our approach provides active security, does not leak any information about the private images, nor about the classifier parameters, and is orders of magnitude more efficient than the privacy-preserving classification solutions proposed in the literature.

Due to their highly accurate predictions, especially for multiclass classification tasks, NNs have superseded SVM as the state-of-the-art for image classification. Nevertheless, our work shows that SVM solutions can be applied efficiently in the privacy-preserving domain. The bottleneck here seems to be the offline phase which can potentially be improved by carefully investigating the parameters for the underlined fixed point precision implementation to minimize the number of triples/bits [6]. Our experiments confirmed that there is a tradeoff between the complexity, and therefore also accuracy of the classification algorithms used, versus the efficiency of the privacy-preserving variants of the proposed solutions. In the active security model that we consider in this work, deploying NNs in the same manner as they are used on cleartext data, is computationally prohibitive with current privacy-preserving methods. An interesting line of research is to investigate whether we can devise privacy-preserving classification algorithms that are as efficient as the SVM, while providing more accurate classification results.

SVM classes	#PCA feat.	#SVs	Accur. (%)	Time online(s)	Time preproc(s)
2	20	128	99.9	0.03	145
3	40	665	99.6	0.16	1107
4	35	1224	99.6	0.22	1631
5	45	1661	99.6	0.33	2700
6	40	2225	99.3	0.42	3089
7	40	2736	99.1	0.50	3751
8	45	3323	98.9	0.59	5099
9	45	4465	98.5	0.84	6617
10	45	5598	98	1.00	8016

Table 2. SVM parameters for best accuracy on MNIST while minimizing the MPC cost. Times given for one SVM evaluation on secret shared HOG features.

References

[1] L. J. Aslett, P. M. Esperança, and C. C. Holmes. Encrypted Statistical Machine Learning: New Privacy Preserving Methods. *arXiv preprint arXiv:1508.06845*, 2015. 4324

[2] A. Barnett, J. Santokhi, M. Simpson, N. P. Smart, C. Stainton-Bygrave, S. Vivek, and A. Waller. Image Classification using non-linear Support Vector Machines on En-

crypted Data. *IACR Cryptology ePrint Archive*, 2017:857, 2017. 4321, 4322, 4324, 4325, 4326, 4327

[3] M. Barni, C. Orlandi, and A. Piva. A Privacy-Preserving Protocol for Neural-Network-Based Computation. In *Proceedings of the 8th workshop on Multimedia and security*, pages 146–151. ACM, 2006. 4321, 4324, 4325

[4] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine Learning Classification over Encrypted Data. In *NDSS*, 2015. 4324, 4325

[5] Bristol Crypto. SPDZ-2: Multiparty Computation with SPDZ Online Phase and MASCOT Offline Phase. https://github.com/bristolcrypto/SPDZ-2, 2016. 4322

[6] O. Catrina and A. Saxena. Secure computation with fixed-point numbers. In *Financial cryptography*, volume 6052, pages 35–50. Springer, 2010. 4328

[7] M. Chase, R. Gilad-Bachrach, K. Laine, K. Lauter, and P. Rindal. Private Collaborative Neural Network Learning. *IACR Cryptology ePrint Archive*, 2017:762, 2017. 4324

[8] M. Dahl. Private Image Analysis with MPC: Training CNNs on Sensitive Data using SPDZ. https://mortendahl.github.io/2017/09/19/private-image-analysis-with-mpc/, 2017. 4324, 4325

[9] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 4325

[10] I. Damgård, K. Damgård, K. Nielsen, P. S. Nordholt, and T. Toft. Confidential Benchmarking based on Multiparty Computation. *IACR Cryptology ePrint Archive*, 2015:1006, 2015. 4323, 4327

[11] I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart. Practical Covertly Secure MPC for Dishonest Majority–or: Breaking the SPDZ Limits. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 2013. 4322, 4323

[12] I. Damgård, V. Pastro, N. Smart, and S. Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In *Advances in Cryptology–CRYPTO 2012*, pages 643–662. Springer, 2012. 4322, 4323

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 4326

[14] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade Object Detection with Deformable Part Models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010. 4325

[15] C. Gentry. Fully Homomorphic Encryption using Ideal Lattices. In *STOC*, volume 9, pages 169–178, 2009. 4324

[16] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. CryptoNets: Applying Neural Networks to Encrypted Data with high Throughput and Accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016. 4321, 4322, 4324, 4325, 4326, 4327

[17] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229. ACM, 1987. 4324

[18] T. Graepel, K. Lauter, and M. Naehrig. ML Confidential: Machine Learning on Encrypted Data. In *International Conference on Information Security and Cryptology*, pages 1–21. Springer, 2012. 4324

[19] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely Connected Convolutional Networks. *arXiv preprint arXiv:1608.06993*, 2016. 4322

[20] J. Jiang and H. Xiong. Fast Pedestrian Detection Based on HOG-PCA and Gentle Adaboost. In *Computer Science & Service System (CSSS), 2012 International Conference on*, pages 1819–1822. IEEE, 2012. 4326

[21] M. Keller, E. Orsini, and P. Scholl. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 830–842. ACM, 2016. 4328

[22] A. Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. 2009. 4327

[23] S. Laur, H. Lipmaa, and T. Mielikäinen. Cryptographically Private Support Vector Machines. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–624. ACM, 2006. 4324, 4325

[24] Y. LeCun, C. Cortes, and C. J. Burges. MNIST Handwritten Digit Database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010. 4327

[25] K.-P. Lin and M.-S. Chen. Privacy-Preserving Outsourcing Support Vector Machines with Random Transformation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 363–372. ACM, 2010. 4324

[26] K.-P. Lin and M.-S. Chen. On the Design and Analysis of the Privacy-Preserving SVM Classifier. *IEEE Transactions on Knowledge and Data Engineering*, 23(11):1704–1717, 2011. 4324

[27] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. In *Advances in CryptologyCRYPTO 2000*, pages 36–54. Springer, 2000. 4324

[28] D. F. Llorca, R. Arroyo, and M. Sotelo. Vehicle Logo Recognition in Traffic Images using HOG Features and SVM. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2229–2234. IEEE, 2013. 4325

[29] W.-L. Lu and J. J. Little. Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor. In *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pages 6–6. IEEE, 2006. 4326

[30] P. Mohassel and Y. Zhang. SecureML: A System for Scalable Privacy-Preserving Machine Learning. *IACR Cryptology ePrint Archive*, 2017:396, 2017. 4322, 4324, 4325

[31] C. Orlandi, A. Piva, and M. Barni. Oblivious Neural Network Computing via Homomorphic Encryption. *EURASIP Journal on Information Security*, 2007:18, 2007. 4324, 4325

[32] Y. Pang, Y. Yuan, X. Li, and J. Pan. Efficient HOG Human Detection. *Signal Processing*, 91(4):773–781, 2011. 4325

[33] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan. Privacy-Preserving Multi-Class Support Vector Machine for Outsourcing the Data Classification in Cloud. *IEEE Transactions on Dependable and Secure Computing*, 11(5):467–479, 2014. 4324, 4325

[34] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On Data Banks and Privacy Homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978. 4324

[35] A. Savakis, R. Sharma, and M. Kumar. Efficient Eye Detection using HOG-PCA Descriptor. *Imaging and Multimedia Analytics in a Web and Mobile World*, 2014:9027, 2014. 4326

[36] R. Shokri and V. Shmatikov. Privacy-Preserving Deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015. 4323, 4324

[37] S. G. Teo, S. Han, and V. C. Lee. Privacy Preserving Support Vector Machine using non-linear Kernels on Hadoop Mahout. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pages 941–948. IEEE, 2013. 4324

[38] J. Vaidya, H. Yu, and X. Jiang. Privacy-Preserving SVM Classification. *Knowledge and Information Systems*, 14(2):161–178, 2008. 4324

[39] V. N. Vapnik and V. Vapnik. *Statistical Learning Theory*, volume 1. Wiley New York, 1998. 4327

[40] A. C.-C. Yao. How to Generate and Exchange Secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986. 4324