MULTI-KERNEL, DEEP NEURAL NETWORK AND HYBRID MODELS FOR PRIVACY PRESERVING MACHINE LEARNING

Mert Al *Thee Chanyaswad* *Sun-Yuan Kung*

Princeton University
Princeton, NJ, USA

ABSTRACT

The rapid rise of IoT and Big Data can facilitate the use of data to enhance our quality of life. However, the omnipresent and sensitive nature of data can simultaneously generate privacy concerns. Hence, there is a strong need to develop techniques that ensure the data serve the intended purposes, but not for prying into one's sensitive information. We address this challenge via utility maximizing lossy compression of data. Our techniques combine the mathematical rigor of Kernel Learning models with the structural richness of Deep Neural Networks, and lead to the novel *Multi-Kernel Learning* and *Hybrid Learning* models. We systematically construct the proposed models in progressive stages, as motivated by the cumulative improvement in the experimental results from the two previously non-intersecting regimes, namely, Kernel Learning and Deep Neural Networks. The final experimental results of the three proposed models on three mobile sensing datasets show that, not only are our methods able to improve the utility prediction accuracies, but they can also cause sensitive predictions to perform nearly as bad as random guessing, resulting in a win-win situation in terms of utility and privacy.

***Index Terms*—** Classification, big data, kernel methods, discriminant information, privacy preserving machine learning

1. INTRODUCTION

With more of our daily activities moving online, a vast amount of personal information is being collected, stored and shared across the internet. Although this information can be used for the benefit of the data owners, it can also leak sensitive information about individuals. Mobile-sensing readings, for instance, can be beneficially used for activity recognition [1], medical diagnosis [2], or authentication [3]; nevertheless, they can also be used to infer sensitive information about individuals such as location, context and identity [4, 5].

The possibility of applying machine learning for adversarial purposes motivates the application of the Principle of Least Privilege to big data [6], i.e., to give users access to only the information necessary for the intended utility, but nothing else. Our methods, hence, follow this principle by seeking the feature representation of the data such that it maximizes the information on the utility task, but removes the rest. We investigate two candidates for this purpose. The first is the *Kernel Based Learning*, which is highly effective for learning low dimensional, utility preserving representations of the data, while the second is *Deep Learning*, which allows us to effectively extract the utility information from multiple feature representations.

Our work is intended to allow privacy preservation to be performed by the data owner even before any information can be extracted. Therefore, we consider two spheres in our design, the *private* and the *public* spheres. From this separation, lossy compression needs to occur in the private sphere, such that any data released to the public sphere should be viable only to the intended purpose.

To achieve such design, we employ compressive encoding schemes that are utility-preserving. Our system is built in progressive stages. We consider the single kernel, multi-kernel, DNN and the hybrid models, and demonstrate the contribution of each to building an effective system for privacy preserving machine learning. We show that Kernel Based Learning successfully removes much of the unnecessary information in data, surpassing standalone Deep Neural Nets (DNNs). However, combining multiple kernels can be necessary to capture all the utility information. This is where DNNs come in, as we show that they can successfully obtain the utility information from multiple kernel embeddings by automatically removing the unnecessary/redundant information.

Finally, we evaluate our models on three mobile sensing datasets. Our multi-kernel models are able to improve activity recognition accuracies from 98.70%, 99.17% and 92.93% to 99.20%, 99.73% and 94.21%, on the three datasets, respectively. Moreover, thanks to the ability of DNNs to distill utility information from multiple kernels, our hybrid model manages to effectively reduce the privacy accuracy to almost random guess, while maintaining high utility performance.

2. RELATED WORK

A number of approaches have been proposed to address the data privacy concern in machine learning. Some of the previous works focus on preserving important statistics of the data, while preventing the original data from being reconstructed [7, 8]. Others directly remove sensitive attributes, which may be redundantly encoded in the data [9, 10]. This work formulates the privacy problem in accordance with the Information Bottleneck Principle, as proposed in [11, 12]. Conceptually, this regime uses compressive encoding to ensure that the published (compressed) data contain only the minimally required information for the intended task, in order to avoid revealing unnecessary information that may be used to infer sensitive (private) information. Particularly, this work considers two classification problems – one for the intended task and another for a sensitive inference. Hence, we employ the *utility maximization data compression* methodology based on the works in [13–15] to ensure that the compressed data can be used to yield high utility classification performance only, but not for any other classification that could leak private information.

3. PRELIMINARIES

Our hybrid model consists of two stages: kernel based compression and deep learning based compression. In the following sections, we describe how kernel based compression is performed, how kernels are selected, and how deep learning based compression is performed.

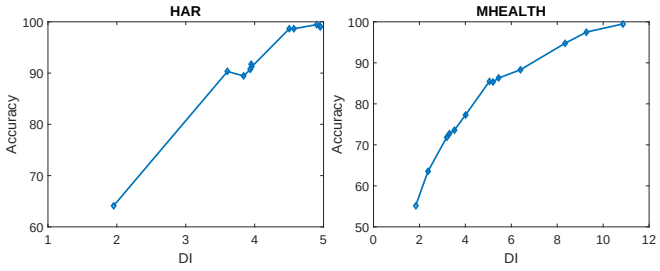


Fig. 1. DI vs. utility accuracies on HAR (left) and MHEALTH (right). Classification accuracies are seen to increase with DI.

3.1. Kernel Based Compression

One regime that has been shown to be effective in the utility-maximizing lossy compression is the kernel-based compression called KDCA [16, 17]. Let $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^J$ be the RKHS mapping for the kernel function. Then $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1) \dots \boldsymbol{\phi}(\mathbf{x}_N)]$ defines the data matrix in the RKHS, and $\bar{\boldsymbol{\Phi}}$ is the center-adjusted data matrix. Then, KDCA searches for the optimal projection matrix $\mathbf{W} \in \mathbb{R}^{J \times m}$:

$$\mathbf{W}_{KDCA} = \underset{\mathbf{W}:\mathbf{W}^T(\bar{\mathbf{S}}+\rho\mathbf{I})\mathbf{W}=\mathbf{I}}{\arg\max} \operatorname{trace}(\mathbf{W}^T\mathbf{S}_B\mathbf{W}), \quad (1)$$

where ρ is a regularization term; $\bar{\mathbf{S}} = \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\Phi}}^T$ is the scatter matrix; and $\mathbf{S}_B = \sum_{c=1}^{L} N_c (\boldsymbol{\mu} - \boldsymbol{\mu}_c)(\boldsymbol{\mu} - \boldsymbol{\mu}_c)^T$ is the between-class scatter matrix, with L being the number of classes, $\boldsymbol{\mu}$ the dataset mean, $\boldsymbol{\mu}_c$ the class mean, and N_c the number of samples in the class c.

Since $\mathbf{W}_{KDCA} \in \operatorname{span}(\bar{\boldsymbol{\Phi}})$ holds [18], we can apply the kernel trick to (1), and the problem becomes that of finding $\mathbf{A} \in \mathbb{R}^{N \times m}$: $\mathbf{W}_{KDCA} = \bar{\boldsymbol{\Phi}}\mathbf{A}$ such that,

$$\mathbf{A}_{KDCA} = \underset{\mathbf{A}:\mathbf{A}^T(\bar{\mathbf{K}}^2+\rho\bar{\mathbf{K}})\mathbf{A}=\mathbf{I}}{\arg\max} \operatorname{trace}(\mathbf{A}^T\mathbf{K}_B\mathbf{A}), \quad (2)$$

where $\bar{\mathbf{K}} = \bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\Phi}}$ is the centered kernel matrix, and $\mathbf{K}_B$ is the kernelized counterpart of $\mathbf{S}_B$ (cf. [17]). Then, the KDCA compression can be applied to the data via the kernel trick:

$$\hat{\boldsymbol{\Phi}} = \mathbf{W}_{KDCA}^T\boldsymbol{\Phi} = \mathbf{A}^T\left(\mathbf{I} - \frac{1}{N}\vec{\mathbf{1}}\vec{\mathbf{1}}^T\right)^T\mathbf{K}. \quad (3)$$

The solution to (2) can be derived via the generalized eigenvalue decomposition (cf. [16]).

We use KDCA to obtain $L-1$ dimensional projections from each kernel induced vector space for two reasons. First, if each class of observations are normally distributed after mapping, with covariance $\bar{\mathbf{S}}$ or $\mathbf{S}_W$, these projections are known to be optimal [19] in the sense that they contain all information $\boldsymbol{\phi}(\mathbf{x})$ has about its label y. Additionally, from the analysis in [15], if the data covariance is represented by its Maximum Likelihood Estimate $\bar{\mathbf{S}}$, these projections capture the maximum mutual information between $\boldsymbol{\phi}(\mathbf{x})$ and the utility subspace spanned by class centers.

3.2. Kernel Selection

It is well known that different kernels provide widely different classification performances for a given task. Hence we perform a filtering procedure based on the Discriminant Information (DI) metric [15, 20]:

$$\mathrm{DI} = \operatorname{trace}\left(\left(\bar{\mathbf{S}} + \rho\mathbf{I}\right)^{-1}\mathbf{S}_B\right) \quad (4)$$

where $\bar{\mathbf{S}}$ and $\mathbf{S}_B$ are defined as in the previous section, and computed using $L-1$ dimensional projections produced by KDCA.

The metric is a measure of the mutual information between the mapping $\boldsymbol{\phi}(\mathbf{x})$ and the label y. Hence removing mappings with low DI score can reduce unnecessary information. This helps regularize the learning space for the DNN [19], which can improve its classification performance. In addition, it reduces the amount of information to be shared, which improves the privacy preservation.

The efficacy of the DI metric can be observed from the accuracies of Support Vector Machines (SVMs) trained using different kernels and tested on our validation sets. As can be observed from Figure 1, an increase in DI generally corresponds to an increase in predictive accuracy. Since KDCA captures the utility information in a kernel induced feature space, and DI selects the best of such spaces, combining KDCA with the kernel selection via DI has an effect of utility-maximizing *space mining*.

3.3. Deep Learning Based Compression

DI based filtering ensures that only the utility information from best feature spaces are considered. However, multiple KDCA representations may contain redundant information with respect to the utility, but such information may inadvertently reveal sensitive information we want to protect. We thus use Deep Neural Networks (DNNs) as another supervised redundancy reduction technique.

Neural Networks can be viewed as methods to learn discriminative, non-linear mappings of the input data. Therefore, to distill utility related information, it is natural to place a narrow, funneling layer that outputs a low-dimensional representation of the original data [21]. For such design, we use fully connected, Feed-Forward Neural Networks with Rectified Linear Units (ReLU) [22], i.e. $f(x) = \max(0, x)$ as activations and a softmax output layer minimizing cross-entropy loss function. Choice of ReLU is motivated by the observation that after KDCA, linear classifiers tend to work as well as non-linear ones.

We borrow a general practice from auto-encoders [21] and use no activations at the narrow hidden layer. This means that non-linear feature mappings are performed up to the narrow layer, which projects these mappings to a subspace. This approach is similar to KDCA subspace projection on the kernel induced space. The difference is that Neural Network learns the mappings together with the projection.

4. METHODOLOGY

In this section, we combine the components described above and propose the novel *Multi-Kernel* and *Hybrid Learning* methods.

4.1. Multi-Kernel Model

We combine the DI filtering method in [20] with the state of the art Multi-Kernel Learning method in [17] to derive our multi-kernel model. Specifically, we select the best kernels via DI filtering, then weight them based on their DI scores. The resulting combined feature representation is used with an SVM. Separation between the KDCA features and the classifier is also the separation between private and public spheres.[1] Comparison with this pure multi-kernel methodology serves to justify the DNN parts of the hybrid models, which we propose next. We call this model **Multi-KDCA** in our experiments.

4.2. Muti-Kernel and DNN Hybrid Models

The multi-kernel method enjoys sound theoretical basis and our experimental results demonstrate that it is effective at removing unneces-

[1] *Our privacy model is to preserve the privacy of test samples. We thus assume training samples to be public or be available to the user.*

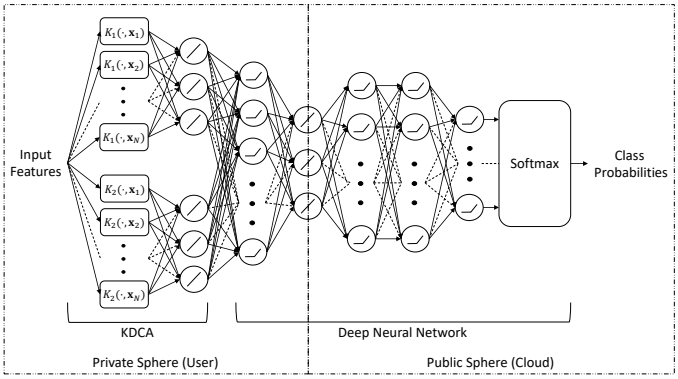


Fig. 2. A schematic of our Compressive Hybrid model. Kernel based projection stage in this example consists of 2 KDCA units projecting to 3 dimensions each, while the Neural Network has a narrow layer with 3 units.

sary information from the data. DNNs on the other hand provide more flexibility and perform well in terms of maintaining utility-related information. Therefore, this motivates us to combine the two models to create the Hybrid Learning model that possesses both advantages.

To combine the strengths of multi-kernel and DNNs, we first perform the space mining on multiple KDCA feature mappings as a feature engineering stage (cf. Section 3.2). Then, we use these mappings as input to the compressive Deep Neural Network (cf. Sec. 3.3). The Neural Network is thus used to discard the redundant/unnecessary information remaining in the multi-kernel projections, as well as to perform classification. To better illustrate the gain from including a DNN with a funneling layer, we consider the two hybrid models:

1. **KDCA+DNN**: This hybrid model uses the multi-kernel features as the input layer of DNN, and then uses a standard feed-forward network with decreasing number of nodes toward the output layer. The separation between the multi-kernel features and the DNN is also that between private and public spheres.
2. **Compressive Hybrid**: This hybrid model again uses the multi-kernel features as the input layer of DNN. However, the Compressive Hybrid model includes a narrow, funneling layer in the DNN architecture (cf. Sec. 3.3). This narrow layer separates the public and private spheres. As is demonstrated in the next section, this hybrid learning model provides the best utility-privacy trade-off among the competing methods. Figure 2 illustrates the architecture of this network.

5. EXPERIMENTS

5.1. Datasets

We evaluate our methods on three mobile sensor readings datasets [23]. In all datasets, we use activity recognition as the utility classification task, and the identity label as a proxy of the unnecessary information encoded in the data. We thus consider person identification as privacy classification. The three datasets used are as follows.

HAR [24] contains 561-feature samples from 30 individuals performing six activities. We randomly partitioned this dataset into training, validation and testing sets with a $5910/1260/1260$ split.

MHEALTH [25] consists of 23-feature data from ten volunteers performing 12 physical activities. The training, validation and testing sets had a $12000/3000/3000$ split.

REALDISP [26] dataset contains 117-feature samples from 17 users performing 33 physical activities. We partitioned the dataset into training, validation and testing sets with a $8000/2000/2000$ split. The sets were obtained from non-overlapping time windows.

We kept the number of users uniformly distributed across datasets in order to have a meaningful comparison with random guessing. If processed data contains no information about the identity of the user, any classifier trained on the data will achieve the same performance as random guessing on the adversarial task of person identification.

5.2. Experimental Setup

We tested our three methods – **Multi-KDCA** (Sec. 4.1), **Multi-KDCA+DNN** (Sec. 4.2) and **Compressive Hybrid** (Sec. 4.2) models – against three benchmark methods including **DNN**, **Compressive DNN** and **Best Single Kernel DCA (KDCA)**. The same DNN network structures are used across our datasets. These were determined via validation using HAR and MHEALTH data.

All of our Neural Networks used ReLU activations (except for the narrow layers with $L-1$ units, which used linear activations), had softmax output layers and minimized cross-entropy loss. Their hidden layer architectures are as follows: **DNNs** had 1024-1024-512-256 units, **Compressive DNNs** had 1024-512-$(L-1)$-256 units, DNN parts of **Multi-KDCA+DNNs** had 1024-1024-512-256 units and DNN parts of **Compressive Hybrids** had 512-$(L-1)$-256-512 units.

For our Single KDCA benchmark, we selected the best kernel via validation. For our Multi-KDCA and Hybrid models, we chose the highest ranking kernels based on their DI scores provided in (4). The number of kernels to be used was determined via validation.

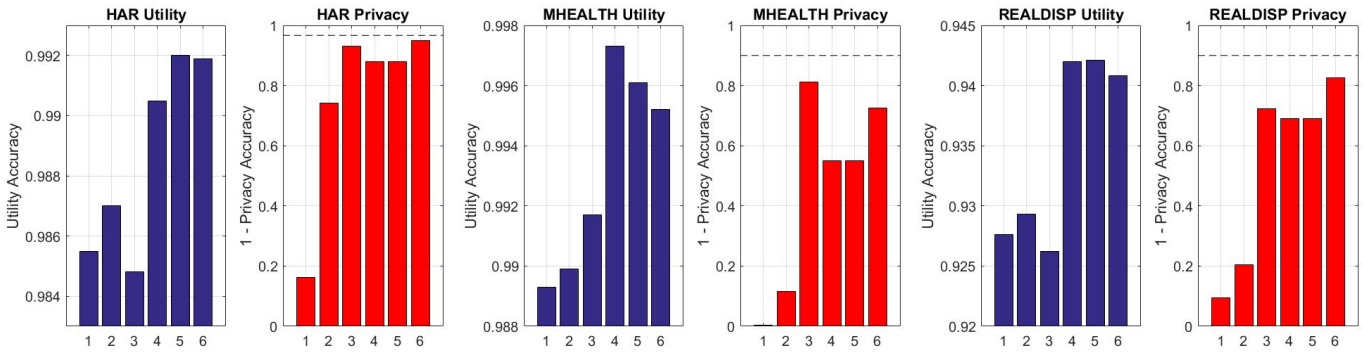
We considered 17 different kernels as follows.

$$
\begin{aligned}
&\text{Linear} && : K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j, \\
&2^{nd} \text{ Degree Poly.} && : K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \gamma \left(\mathbf{x}_i^T \mathbf{x}_j\right)\right)^2, \\
&3^{rd} \text{ Degree Poly.} && : K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \gamma \left(\mathbf{x}_i^T \mathbf{x}_j\right)\right)^3, \\
&\text{Laplacian} && : K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_1\right), \\
&\text{RBF} && : K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right),
\end{aligned}
$$

with $\gamma \in \{1, 0.1, 0.01, 0.001\}$. After DI score based filtering by monitoring validation accuracies, we ended up using 9 kernels on HAR, 4 kernels on MHEALTH and 5 kernels on REALDISP datasets for our Multi-KDCA and Hybrid models. We set the hyper-parameter ρ to 0.01 for both KDCA (2) and the DI metric (4).

We used the Adam Optimizer [27] with batch size 50 for training the Neural Networks. We set step size to $\alpha = 0.0001$, first moment smoothing factor to $\beta_1 = 0.9$ and second moment smoothing factor to $\beta_2 = 0.999$. In order to regularize the Neural Networks, we applied drop-out to all hidden layers except the linear ones with width $L-1$. The probability of dropping a unit was set to 0.5. In addition, we used early stopping on validation sets to prevent overfitting.

To test the privacy performances of our models against the person identification task, we used 5 different Neural Networks, in addition to SVMs with RBF kernels, due to the tendency of the former to overfit the privacy task on compressed data. We used DNNs with 128-128, 256-256, 512-512, 1024-1024, 512-512-128 hidden layer architectures and SVMs against the Compressive DNN, Single Kernel DCA, Multi-Kernel DCA and Hybrid models. Against the DNN model, we used a 1024-1024-512-256 hidden layer network and SVMs with RBF kernels.



Methods: 1: DNN, 2: Compressive DNN, 3: Best Single KDCA, **4: Multi-KDCA**, **5: Multi-KDCA+DNN**, **6: Compressive Hybrid**

Fig. 3. The utility and privacy performances on HAR (left), MHEALTH (middle) and REALDISP (right). Dashed line represents the best privacy performance achievable, which corresponds to random guess. **HAR**: Multi-KDCA+DNN and Compressive Hybrid achieve the best utility, Compressive Hybrid also achieves the best privacy. **MHEALTH**: Multi-KDCA achieves the best utility, Compressive Hybrid achieves significantly better privacy with high utility, Single KDCA achieves the best privacy. **REALDISP**: Multi-KDCA and Multi-KDCA+DNN achieve the best utility and Compressive DNN achieves the best privacy with similar utility.

5.3. Results

The results are reported in Figure 3. Here are key observations.

5.3.1. HAR

Our **Multi-KDCA**, **Multi-KDCA+DNN** and **Compressive Hybrid** models achieve 99.05%, 99.20% and 99.19% utility classification accuracies, respectively. We outperform all three benchmark methods, the best of which has the accuracy of 98.70%.

Among the benchmark methods, **Single KDCA** achieves the best privacy accuracy 5.33% (note that the plots show $1-$ privacy accuracy). Although our **Multi-KDCA** and **Multi-KDCA+DNN** models have slightly worse privacy performance, both with 12.06%, they provide considerable gain in utility. Our **Compressive Hybrid**, on the other hand, achieves better privacy performance than all benchmark methods with 5.06%, which is close to random guess at 3.33%. This reinforces the effectiveness of DNN at removing unnecessary information from multiple kernels.

5.3.2. MHEALTH

Our **Multi-KDCA**, **Multi-KDCA+DNN** and **Compressive Hybrid** models achieve 99.73%, 99.61% and 99.52% utility classification accuracies, respectively. We outperform all three benchmark methods, the best of which has the accuracy of 99.17%.

For privacy, **Single KDCA** proves the most effective with 18.80% privacy accuracy. Our **Multi-KDCA** and **Multi-KDCA +DNN** models, however, have less impressive privacy performance at 44.97% accuracy. Our **Compressive Hybrid**, nonetheless, is relatively effective in the privacy preservation with 27.38% accuracy. Here, we observe that the **Single KDCA** performs best for the privacy task, despite its lower utility performance. This implies a significant overlap between utility and privacy information on this dataset.

5.3.3. REALDISP

Our **Multi-KDCA**, **Multi-KDCA+DNN** and **Compressive Hybrid** models achieve 94.21%, 94.20% and 94.08% utility classification accuracies, respectively. We outperform all three benchmark methods, the best of which has the accuracy of 92.93%.

For privacy, **Single KDCA** achieves the best performance among the three benchmark methods at 27.75% accuracy. Our **Multi-KDCA** and **Multi-KDCA+DNN** also perform well, both at 30.85% accuracy. Furthermore, our **Compressive Hybrid** model is the most successful at removing sensitive information here, with 17.46% accuracy, which is close to random guess at 10%. Thus including a DNN with a funneling layer proves effective at removing additional information, while maintaining the same utility as the Multi-Kernel models.

5.3.4. Summary

A consistent observation across all datasets is that utility is improved when utility information from multiple kernels are combined. **Multi-KDCA** and **Multi-KDCA+DNN** achieve the best utility performances, while they also remove identity information better than Compressive DNNs. This reinforces the importance of the space mining process. **Multi-KDCA+DNNs** without kernel weighting also perform as well as kernel weighted **Multi-KDCAs**, showing that DNNs can automatically and effectively learn the kernel weights.

Compressive Hybrid achieves approximately the same utility performance as **Multi-KDCAs** and **Multi-KDCA+DNNs**. However, it achieves significantly better privacy than other models. This is primarily due to the narrow funneling layer, in addition to the kernel based compression and space mining. Compressive Hybrid thus proves suitable for removing redundancies contained in multiple kernel embeddings.

6. CONCLUSION

We proposed novel multi-kernel and hybrid methods in order to remove unnecessary information from the data, while maintaining utility-related information. Results we obtained from three datasets have shown that, our models can significantly improve utility performance, while successfully reducing sensitive information contained in the data. Additionally, our work establishes a suitable framework for combining the strengths of Kernel Learning and Deep Learning. Future work may improve the scalability of our approach and extend our hybrid methodology to other types of network architectures.

Acknowledgments

This material is based on work supported in part by the Brandeis Program of the Defense Advanced Research Project Agency (DARPA) and Space and Naval Warfare System Center Pacific (SSC Pacific) under Contract No. 66001-15-C-4068. We thank Prof. J. Morris Chang and Prof. Pei-Yuan Wu for making this research possible.

7. REFERENCES

[1] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[2] E. Agu, P. Pedersen, D. Strong, B. Tulu, Q. He, L. Wang, and Y. Li, "The smartphone as a medical device: Assessing enablers, benefits and challenges," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pp. 76–80, IEEE, 2013.

[3] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pp. 1–7, IEEE, 2010.

[4] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 397–413, IEEE, 2016.

[5] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.

[6] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.

[7] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Transactions on knowledge and Data Engineering*, vol. 18, no. 1, pp. 92–106, 2006.

[8] B. Liu, Y. Jiang, F. Sha, and R. Govindan, "Cloud-enabled privacy-preserving collaborative learning for mobile sensing," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pp. 57–70, ACM, 2012.

[9] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning*, pp. 325–333, 2013.

[10] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[11] S. Kung, "A compressive privacy approach to generalized information bottleneck and privacy funnel problems," *Journal of the Franklin Institute*, 2017.

[12] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proceedings of the 37th Allerton Conference on Communication, Control and Computing*, pp. 368–377, University of Illinois, 1999.

[13] S.-Y. Kung, T. Chanyaswad, J. M. Chang, and P. Wu, "Collaborative pca/dca learning methods for compressive privacy," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, p. 76, 2017.

[14] T. Chanyaswad, J. M. Chang, P. Mittal, and S. Kung, "Discriminant-component eigenfaces for privacy-preserving face recognition," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pp. 1–6, IEEE, 2016.

[15] S.-Y. Kung, "Compressive privacy: From information/estimation theory to machine learning [lecture notes]," *IEEE Signal Processing Magazine*, vol. 34, no. 1, pp. 94–112, 2017.

[16] S.-Y. Kung, "Discriminant component analysis for privacy protection and visualization of big data," *Multimedia Tools and Applications*, pp. 1–36, 2015.

[17] T. Chanyaswad, J. M. Chang, and S. Y. Kung, "A compressive multi-kernel method for privacy-preserving machine learning," in *International Joint Conference on Neural Networks (IJCNN), 2017*, IEEE, 2017.

[18] S. Y. Kung, *Kernel methods and machine learning*. Cambridge University Press, 2014.

[19] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin, 2001.

[20] T. Chanyaswad, M. Al, J. M. Chang, and S. Y. Kung, "Differential mutual information forward search for multi-kernel discriminant-component selection with an application to privacy-preserving classification," in *Machine Learning for Signal Processing (MLSP), 2017 IEEE 27th International Workshop on*, IEEE, 2017.

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

[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[23] K. Bache and M. Lichman, "Uci machine learning repository," 2013.

[24] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones.," in *ESANN*, 2013.

[25] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International Workshop on Ambient Assisted Living*, pp. 91–98, Springer, 2014.

[26] O. Baños, M. Damas, H. Pomares, I. Rojas, M. A. Tóth, and O. Amft, "A benchmark dataset to evaluate sensor displacement in activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 1026–1035, ACM, 2012.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.