References

1 Yang, M., et al.: Denseaspp for semantic segmentation in street scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3684–3692. IEEE, Piscataway, NJ (2018)

2 Chen, L.-C., et al.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2017)

3 Liu, Z., Qi, X., Torr, P.H.: Global texture enhancement for fake face detection in the wild. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8060–8069. IEEE, Piscataway, NJ (2020)

4 Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423. IEEE, Piscataway, NJ (2016)

5 Hu, J., Yamasaki, T., Aizawa, K.: Multimodal learning for image popularity prediction on social media. *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1–2. IEEE, Piscataway, NJ (2016)

6 Zhang, Z., et al.: How to become instagram famous: Post popularity prediction with dual-attention. *2018 IEEE International Conference on Big Data (Big Data)*. pp. 2383–2392. IEEE, Piscataway, NJ (2018)

7 Zhang, W., et al.: User-guided hierarchical attention network for multimodal social image popularity prediction. *Proceedings of the 2018 World Wide Web Conference*, pp. 1277–1286. ACM, New York, NY (2018)

Privacy-preserving evaluation for support vector clustering

J. Byun,[1] J. Lee,[1] and S. Park[2,✉]

[1]*Department of Industrial Engineering, Seoul National University, Korea*

[2]*Department of Convergence Security Engineering, Sungshin University, Korea*

✉E-mail: psr6275@sungshin.ac.kr

The authors proposed a privacy-preserving evaluation algorithm for support vector clustering with a fully homomorphic encryption. The proposed method assigns clustering labels to encrypted test data with an encrypted support function. This method inherits the advantageous properties of support vector clustering, which is naturally inductive to cluster new test data from complex distributions. The authors efficiently implemented the proposed method with elaborate packing of the plaintexts and avoiding non-polynomial operations that are not friendly to homomorphic encryption. These experimental results showed that the proposed model is effective in terms of clustering performance and has robustness against the error that occurs from homomorphic evaluation and approximate operations.

Introduction: Recently, machine learning technologies have successfully solved real-world problems in various fields, including biomedical and financial applications. Sensitive data such as personal data, biometric data, medical information, and financial information can be used to construct high-quality machine learning models. Therefore, moral and legal issues about data protection and privacy use have received attention, and the conflict between data utilization and protection needs to be addressed.

Fully homomorphic encryption (FHE), which enables numerical operations on encrypted data, is considered to be a promising direction that satisfies data utilization and protection [1, 2]. Privacy-preserving machine learning algorithms with FHE have been proposed to train supervised models and evaluate the models on the encrypted domain, where encrypted data can be transmitted to the model without revealing the original data [3, 4]. However, implementing machine learning algorithms with FHE involves much slower computations and much larger data storage than implementing the same algorithms on the plaintext domain. Implementing machine learning algorithms without considering the operational characteristics of FHE can worsen the problems and degrade the performance.

Clustering is a representative unsupervised learning task widely used in areas including image segmentation, information retrieval, and marketing. Clustering algorithms partition given instances into a set of subgroups called clusters depending on their similarity (or distance). Clustering on the encrypted data can be more complicated than classification or regression because the shape and the number of clusters are unknown. Clustering algorithms such as k-means clustering and mean shift clustering methods have been implemented on encrypted data but lack the performance of complex and non-convex data [5, 6]. In contrast, the support vector clustering (SVC) algorithm can capture the complex shape of clusters by labelling the support of data distribution based on the support vector domain description (SVDD) [7, 8].

In this letter, we propose a privacy-preserving evaluation for SVC. Our work aims to implement an efficient SVC inference on the encrypted domain, where it can capture the complex data distribution with a support function and assign the most appropriate cluster for a new test data. Our algorithm enables robust SVC labelling for test data on an encrypted domain without decryption by configuring the entire procedure as a homomorphic operation. In the experiments, six datasets were used to evaluate the performance of clustering algorithms on the encrypted domains.

FHE and HEAAN scheme: An FHE scheme aims to construct a homomorphic encryption (HE) scheme that supports an unbounded number of operations for evaluating any function f on encrypted data without decryption [1]. An FHE scheme consists of four procedures: KeyGen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$ and $\mathsf{Evaluate}$. The correctness condition for evaluated ciphertexts ensures that for any function f, a plaintext m and its encryption c, the decryption result of the ciphertext $c' = \mathsf{Evaluate}(f, c)$ is the same as $f(m)$. In FHE, any evaluation function is represented as a composition of homomorphic additions and multiplications. Thus, homomorphic evaluations of non-polynomial operations can consume a much longer time and have faster error propagation than the evaluations on the plaintext domain.

In this letter, we used a HEAAN scheme that supports the approximate computation of real numbers, where a small error is added to the plaintext vector after decryption [2]. HEAAN can provide efficient floating-point operations at the expense of a bounded loss of precision. A complex vector (plaintext) can be encoded into a ring element and encrypted into a single ciphertext, and the slot rotation of the vector on the encrypted domain enables an efficient parallel computation of the ciphertexts. Because of efficient computation, many machine learning algorithms based on HEAAN have been proposed for real-world applications [4, 6].

Initializing the HEAAN scheme, some parameters are determined to achieve a targeted level of security, including N, initial ciphertext modulus $\log q_L$ and scaling factor p, where N is related to the ciphertext space and the number of plaintext slots ($= N/2$). The parameters $\log q_L$ and p determine the precision of the calculations and the number of operations without bootstrapping. Since the error in the evaluated ciphertext grow rapidly with multiplication compared to other operations, it is necessary to rescale the ciphertext after multiplication to control the magnitude of the error with decreasing the ciphertext modulus.

The bootstrapping procedure allows the evaluated ciphertext to be refreshed by homomorphically evaluating the $\mathsf{Decrypt}$ function with increasing the ciphertext modulus. In the case of HEAAN, the computational cost of bootstrapping increases with the number of plaintext slots with $O(\log N/2)$ [9]. Although bootstrapping of HEAAN is more efficient than other FHE schemes, it is still the most expensive part of HEAAN. Therefore, it is essential to consider the trade-off between the efficiency of parallel operation and the cost of bootstrapping. For detailed information about the scheme, we refer the readers to [2, 9].

SVC: Support-based clustering starts with estimating the support function of data distribution obtained by the SVDD, Gaussian process clustering, or kernel density estimation. In this study, we used SVDD with the Gaussian kernel to obtain the support function as follows:

$$s(\mathbf{x}) = 1 - 2\sum_{i=1}^{N_v} \beta_i e^{-\delta\|\mathbf{x}-\mathbf{x}_i\|^2} + \sum_{i=1}^{N_v}\sum_{j=1}^{N_v} \beta_i\beta_j e^{-\delta\|\mathbf{x}_i-\mathbf{x}_j\|^2}, \qquad (1)$$

1350911x, 2021, 2, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.12047 by Chung-Ang University, Wiley Online Library on [13/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ALGORITHM 1 HE friendly evaluation of support-based clustering

```
Input: Test data $\{\mathbf{x}_i\}_{i=1}^{N_{test}}$, SEVs $\{\mathbf{s}_i\}_{i=1}^{N_S}$, SVs $\{\mathbf{v}_i\}_{i=1}^{N_V}$, support function $s$, number of iteration $T$, learning rate $\eta$
Output: Clustering label $\{l_i\}_{i=1}^{N_{test}}$
for $i = 1$ to $N_{test}$ do
    $\mathbf{x}_i^0 \leftarrow \mathbf{x}_i$
    for $t = 1$ to $T$ do
        $\mathbf{x}_i^t \leftarrow \mathbf{x}_i^{t-1} - \eta \cdot \nabla s(\mathbf{x}_i^{t-1})$
    end for
    Find the nearest SEV ($\mathbf{s}_i$) of $\mathbf{x}_i^T$, and label $\mathbf{x}_i$ as $l_i$, the label of $\mathbf{s}_i$.
end for
```

where $\delta > 0$ is the width parameter for the Gaussian kernel and N_v denotes the number of support vectors (SVs). The support function (1) can be used to partition the data space into basin cells by constructing the following dynamical system:

$$\frac{d\mathbf{x}}{dt} = -\nabla s(\mathbf{x}). \tag{2}$$

A stable equilibrium vector (SEV) of the system (2) is an equilibrium state where all the eigenvalues of Hessian $\nabla^2 s(\mathbf{x})$ are positive. Then, the basin cell $B(\mathbf{s}_i)$ is defined as the closure of the set of all the data points that converge to a SEV $\mathbf{s}_i$ following the system (2). Some SEVs are connected to constitute clusters using the characteristic of the dynamical system (2), and the points in a basin cell are labelled with the cluster label of the corresponding SEV [7, 8, 10]. SVC can be inductive and have an efficient and stable inference phase using the system (2) [7, 10].

HE-friendly evaluation of SVC: In user-server scenarios, the inference phase of the clustering algorithm with FHE can provide to partition the given encrypted instances while protecting user's privacy from curious servers. In this letter, we present the inference phase of the SVC algorithm with FHE because SVC is naturally inductive and has a stable inference. For SVC, most inference methods are based on the system (2) that a test point follows to find the SEV. In particular, the simplest inference method is to allocate the test points to the cluster of the closest SEV. However, this inference cannot capture the complex clusters because it ignores the intrinsic data distribution. Therefore, we present an elaborate inference algorithm of SVC which utilizes the support function (1) of the data distribution. We addressed the challenge of dealing with HE-unfriendly operations of SVC inference while balancing efficiency and accuracy.

We propose a HE-friendly evaluation that utilizes the basin induced from the system (1) to stabilize the inference. Our algorithm consists of two parts. In the first part, we use the dynamical system (2) to move the given test data to more stable points. We can postulate that the boundary regions between two different basins are the most unstable, whereas the SEVs are the most stable points. Thus, our algorithm makes the points evade the boundary region by applying the system (2) which converges to SEVs. In the second part, our algorithm assigns the cluster label by finding the SEV closest to the point obtained after the first part. The procedure of our proposed method is presented in Algorithm 1.

In Algorithm 1, we need to implement the calculation of the gradient of the support function $\nabla s(\mathbf{x})$ homomorphically. When using the Gaussian kernel, $\nabla s(\mathbf{x})$ can be directly calculated as:

$$\nabla s(\mathbf{x}) = 4\delta \sum_{j=1}^{N_v} \beta_j e^{-\delta\|\mathbf{x}-\mathbf{v}_j\|^2} \cdot (\mathbf{x} - \mathbf{v}_j). \tag{3}$$

HEAAN is used to efficiently implement the approximate computation of real numbers on encrypted data, which supports homomorphic additions and multiplications. However, it requires polynomial approximations for non-polynomial operations. The gradient (3) contains the exponential function and the distance between a test point and the SVs.

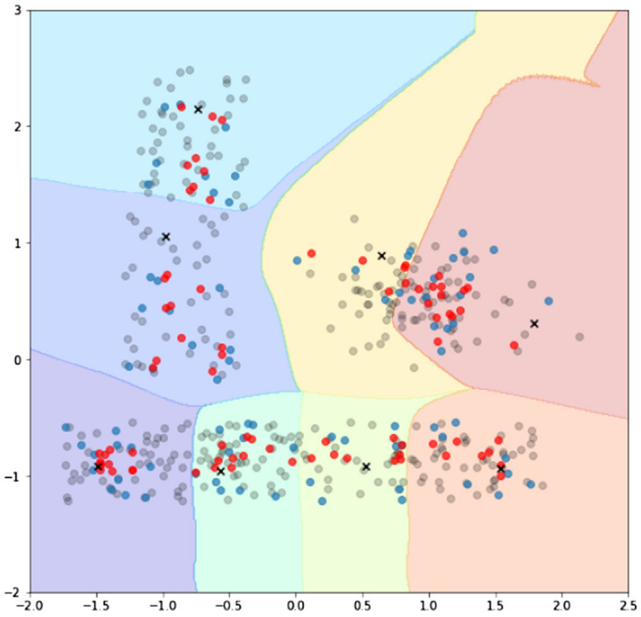


Fig. 1 *Illustration of gradient step for Algorithm 1, where the gray points represent training data, the blue points represent test data, the red points represent the data point after each gradient step, and each "x" point represents the corresponding SEV of the basins region with different colors*

After the first part has been found without decryption, finding the nearest SEV consists of two components: calculating the distances between a test point and the SEVs and finding the minimum distance between them. The exponential function is approximated with a Taylor expansion. To obtain the min-index of the distances, we used the iterative algorithm in [11]. However, the appropriate packing strategy is needed to improve the computational efficiency of calculating the distances and finding the min-index.

We design the plaintext packing to avoid bootstrapping. For simplicity, we assume that all vectors, including test data, SVs, and SEVs, are d-dimensional row vectors. Figure 2 shows the structures of the packed plaintexts, where a plaintext vector will be encrypted into a ciphertext. Note that HEAAN supports addition, element-wise multiplication, and right-shift and left-shift rotations for plaintext vectors on the encrypted domain. We designed the plaintext vector to efficiently calculate the distances using these operations because both parts of Algorithm 1 involve the calculation of the distances. The test samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{test}}$ are repeated as many as N_V, while each instance of SVs $\mathbf{v}_i$ and SEVs $\mathbf{s}_i$ are repeated as many as the number of test data N_{test} as in Figure 2. For SEVs, N_{test} times repeated vectors are additionally repeated N_V/N_S times. The plaintexts become $dN_{test}N_V$-dimensional vectors, which consist of d-dimensional sub-vectors. In this way, subtracting all SVs or SEVs from all test data can be done with only one operation. In addition, $\beta = [\beta_1, \ldots, \beta_{N_V}]$ were packed in the same way as SVs, where β_j's were repeated d times to constitute a vector like $\mathbf{v}_j \in \mathbb{R}^d$. Depending on the number of test samples N_{test} and SVs N_V, we can use multiple ciphertexts for a plaintext vector and parallelize the homomorphic operations.

Finally, we can efficiently compute the difference $\mathbf{x}_i - \mathbf{v}_j$ for obtaining the gradient (3) and $\mathbf{x}_i - \mathbf{s}_j$ for finding the closest SEV. After constituting all necessary ciphertexts, we can compute the gradient using homomorphic operations without decryption by replacing the entire operations with the polynomial operations. When finding the nearest SEV, we use the min-index algorithm whose output has a reciprocal of the number of minimal elements for the indices of the minimal elements and 0 for the other indices as in [6]. Because the min-index algorithm on the encrypted domain involves an approximate error, the outputs of the minimal elements can be indistinguishable from 0s with the errors when the number of minimal elements is large. However, the gradient phase of our algorithm induces the test samples near the boundary to move to the SEV, resulting in more accurate homomorphic results for the min-index algorithm. Figure 1 illustrates the change of test points after one gradient

1350911x, 2021, 2, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.12047 by Chung-Ang University, Wiley Online Library on [13/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

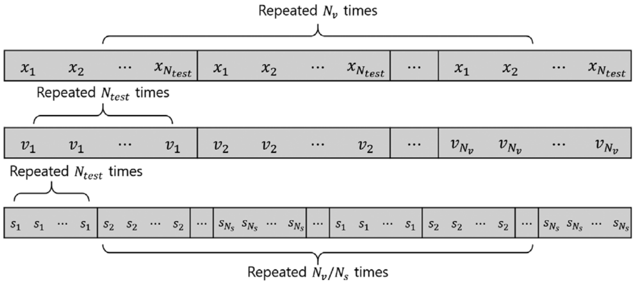

Repeated N_v times
x_1 x_2 ⋯ $x_{N_{test}}$ x_1 x_2 ⋯ $x_{N_{test}}$ ⋯ x_1 x_2 ⋯ $x_{N_{test}}$
Repeated N_{test} times
v_1 v_1 ⋯ v_1 v_2 v_2 ⋯ v_2 ⋯ v_{N_v} v_{N_v} ⋯ v_{N_v}
Repeated N_{test} times
s_1 s_1 ⋯ s_1 s_2 s_2 ⋯ s_2 ⋯ s_{N_s} s_{N_s} ⋯ s_{N_s} ⋯ s_1 s_1 ⋯ s_1 s_2 s_2 ⋯ s_2 ⋯ s_{N_s} s_{N_s} ⋯ s_{N_s}
Repeated N_v/N_s times

Fig. 2 *Description of the packing method*

Table 1. Summarization of the datasets

Dataset	Instances	Attributes	Clusters	Convexity
Hepta	212	3	7	convex
Tetra	400	3	4	convex
Lsun	400	2	3	convex
Two diamonds	800	2	2	convex
Target	758	2	2	non-convex
Chainlink	1000	3	2	non-convex

descent iteration. We can notice that the test points move toward the SEVs of their basins and away from the other SEVs simultaneously. It can affect the clustering performance for complex data distribution.

Experiments: We evaluated the proposed method on six datasets shown in Table 1 from the fundamental clustering problems suite [12]. We compared clustering performance in terms of the adjusted Rand index (ARI) metric and computation time with HE-friendly mean shift clustering (Meanshift) and k-means clustering (KMeans), which are currently the state-of-the-art clustering models used with FHE. The ARI measures the similarity between two data partitions and has a value between 0 and 1, where 1 represents a perfect agreement between two data partitions. To verify how the error included in ciphertext affects the clustering results, we compared the results of Algorithm 1 to encrypted data (ARI-enc) and unencrypted data (ARI-no). We implemented the evaluation of KMeans and Meanshift, following [6].

We used an Intel Xeon CPU E5-2660 v3 @2.60GHZ processor. We set $\log q_L = 1200$, $\log p = 30$, $N = 2^{16}$ and $\delta = 2$ for all datasets. We conducted the experiments with different test rates in $\{0.2, 0.5, 0.9\}$, which means the ratio between the number of test data and the number of training and test data. Training data was sampled five times for each experiment to measure the average performance. For Algorithm 1, we used one iteration step of (2), where the learning rate was set to 0.5 for the Chainlink dataset and 0.8 for the other datasets.

Table 2 shows the clustering results. KMeans showed almost the same computation time for all experiments since a plaintext vector is encrypted into a single ciphertext. In contrast, except Hepta and Lsun, the proposed method showed the longest execution time because it needed to split a plaintext into multiple ciphertexts. The total execution time is highly dependent on the number of ciphertexts, which is related to the number of SVs for the proposed method, to the number of modes for Meanshift, and to the number of clusters for Kmeans. Therefore, we can reduce the computational cost if the SVC algorithm obtains a sparse support function (1) with few support vectors.

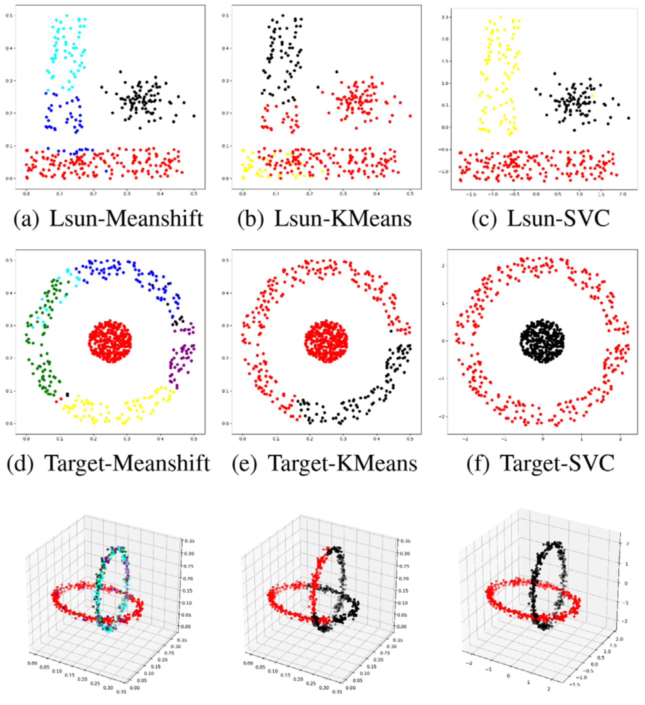
For Hepta and Tetra datasets, which have simple and convex distributions, all the models showed good performance. For all the other datasets, our method showed the highest ARI value regardless of the test rate. The other algorithms showed poor performance, especially for the Target and Chainlink datasets that have non-convex data distributions, while our model achieved high ARIs. For the Lsun dataset that consists of rectangular clusters with different lengths and widths, the ARI values of the KMeans and Meanshift algorithms for encrypted and unencrypted data are different significantly, whereas SVC always showed consistent results even after encryption. These results demonstrate that by pulling the data point at the boundary toward the center of the basin cell, SVC can attain robustness against the error that can occur from homomorphic operations.

Table 2. Comparison of the results of three algorithms on FCPS datasets. For each experiment, the highest ARI value is highlighted in bold

Dataset	Test rate	KMeans		Meanshift		SVC	
		ARI-enc	time (s)	ARI-enc	time (s)	ARI-enc	time (s)
		ARI-no		ARI-no		ARI-no	
Hepta	0.2	1	16.967	1	17.332	1	30.402
		1		1		1	
	0.5	1	17.001	1	16.994	1	28.360
		1		1		1	
	0.9	1	17.025	1	17.028	1	27.239
		1		1		1	
Tetra	0.2	**1**	16.162	0.957	16.962	0.927	57.585
		1		0.970		0.927	
	0.5	**1**	16.505	0.973	17.383	0.947	113.896
		1		0.973		0.947	
	0.9	**0.994**	16.110	0.915	17.033	0.961	109.437
		0.994		0.915		0.961	
Lsun	0.2	0.351	15.386	0.629	16.543	**0.875**	25.673
		0.417		0.613		**0.875**	
	0.5	0.346	15.328	0.611	16.494	**0.931**	25.837
		0.400		0.651		**0.931**	
	0.9	0.342	15.409	0.663	16.171	**0.993**	25.368
		0.400		0.734		**0.993**	
Two diamonds	0.2	0.897	14.978	0.812	15.476	**0.995**	26.147
		0.897		0.812		**0.995**	
	0.5	0.891	15.085	0.842	15.404	**0.934**	52.469
		0.891		0.842		**0.934**	
	0.9	0.879	14.640	0.777	15.534	**0.923**	102.129
		0.879		0.777		**0.923**	
Target	0.2	0.105	14.694	0.619	17.055	**1**	26.375
		0.105		0.623		**1**	
	0.5	0.119	14.822	0.620	17.331	**1**	53.773
		0.119		0.620		**1**	
	0.9	0.121	14.857	0.627	17.185	**1**	104.359
		0.121		0.627		**1**	
Chain-link	0.2	0.094	15.639	0.235	18.890	**0.853**	111.958
		0.094		0.231		**0.853**	
	0.5	0.098	15.465	0.228	37.062	**0.887**	223.896
		0.098		0.229		**0.887**	
	0.9	0.068	15.548	0.318	15.513	**0.967**	220.380
		0.068		0.318		**0.967**	

Figure 3 shows the result of our investigation of the difference in clustering performance by presenting the clustering results for three data sets. Figure 3a,b illustrates that KMeans and Meanshift failed to capture the clusters, especially for the data points at the boundary of the divided region. However, our method correctly allocated the clusters because of the robustness of our method, as stated in the above paragraph.

1350911x, 2021, 2, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.12047 by Chung-Ang University, Wiley Online Library on [13/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Lsun-Meanshift (b) Lsun-KMeans (c) Lsun-SVC

(d) Target-Meanshift (e) Target-KMeans (f) Target-SVC

(g) Chain-Meanshift (h) Chain-KMeans (i) Chain-SVC

Fig. 3 *Visualization of clustering results for Lsun, Target and Chainlink datasets. Each estimated cluster is color-coded*

In the case of datasets with non-convex distributions such as Target and Chainlink, we found that KMeans essentially did not reflect the non-convex distribution, and Meanshift was not able to connect the clusters into a single cluster. On the other hand, SVC completely clustered the Target dataset, and for the Chainlink dataset, almost all data points were properly clustered except for the points located where two clusters are adjacent to each other.

Conclusion: In this letter, we proposed a privacy-preserving evaluation algorithm of SVC with fully homomorphic encryption. Our model enables the allocation of the cluster label for new test data without decryption, improving the clustering performance for non-convex data, and provides robustness of data points near the boundary. The experimental results show that the proposed method effectively clusters encrypted data with various distributions in a realistic amount of time. In the future, we can improve the computational cost of our algorithm by training a sparse SVC model and parallelizing computationally expensive operations when using multiple ciphertexts.

Acknowledgements: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1A02085851), and in part by the NRF Grant funded by the Korean Government (MSIT) (NRF-2019R1A2C2002358).

© 2020 The Authors. *Electronics Letters* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Received: *23 September 2020* Accepted: *27 October 2020*
doi: 10.1049/ell2.12047

References

1 Gentry, C.: Fully homomorphic encryption using ideal lattices. In: *Proceedings of the 41st ACM symposium on Theory of Computing–STOC*, pp. 169–178. ACM, New York (2009)

2 Cheon, J.H., et al.: Homomorphic encryption for arithmetic of approximate numbers. In: *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437. Springer, Cham (2017)

3 Graepel, T., Lauter, K., Naehrig, M.: ML confidential: Machine learning on encrypted data. In: *International Conference on Information Security and Cryptology*, pp. 1–21. Berlin, Heidelberg (2012)

4 Park, S., et al.: HE-friendly algorithm for privacy-preserving SVM training. *IEEE Access* **8**, 57414–57425 (2020)

5 Almutairi, N., Coenen, F., Dures, K.: K-means clustering using homomorphic encryption and an updatable distance matrix: Secure third party data clustering with limited data owner interaction. In: *International Conference on Big Data Analytics and Knowledge Discovery*, pp. 274–285. Springer, Cham (2017)

6 Cheon, J.H., Kim, D., Park, J.H.: Towards a practical cluster analysis over encrypted data. In: *International Conference on Selected Areas in Cryptography*, pp. 227–249. Springer, Cham (2019)

7 Lee, J., Lee, D.: Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1869–1874 (2006)

8 Kim, K., Son, Y., Lee, J.: Voronoi cell-based clustering using a kernel support. *IEEE Trans. Knowledge Data Eng.* **27**(4), 1146–1156 (2014)

9 Cheon, J.H., et al.: Bootstrapping for approximate homomorphic encryption. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 360–384. Springer, Cham (2018)

10 Park, S., Hah, J., Lee, J.: Inductive ensemble clustering using kernel support matching. *Electron. Letters* **53**(25), 1625–1626 (2017)

11 Cheon, J.H., et al.: Numerical method for comparison on homomorphically encrypted numbers. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, Cham (2019)

12 Ultsch, A.: Fundamental clustering problems suite (FCPS). Technical report, University of Marburg (2005)

Enhancement of sensing characteristics of Polydimethylsiloxane-based capacitive force sensor by introducing conductive polymer to dielectric layer

Yasumin Siangkhio,[1] Adirek Rangkasikorn,[1]
Narin Tammarugwattana,[2] Navaphun Kayunkid,[1,✉]
Sukittaya Jessadaluk,[1] Sakon Rahong,[1]
Supamas Wirunchit,[1] and Jiti Nukeaw[1]
[1]*College of Nanotechnology, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, Thailand*
[2]*Faculty of Engineering, Department of Instrumentation and Control Engineering, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, Thailand*
✉Email: navaphun.ka@kmitl.ac.th

A capacitive force sensor is one of the electronics components used in several electronic devices and applications. An improvement of sensing characteristics of the sensor, for example sensitivity and response time, becomes an interesting research topic. The alternative approach to enhance the sensitivity and response time of polydimethylsiloxane-based capacitive force sensors is proposed by introducing poly(3,4-ethylenedioxythiophene) polystyrene sulphonate, a conductive polymer, into polydimethylsiloxane active layer. Two sensors using different active layers, (i) polydimethylsiloxane (conventional sensor) and (ii) poly(3,4-ethylenedioxythiophene) polystyrene sulphonate mixed polydimethylsiloxane (modified sensor), were fabricated and characterised to reveal the sensing enhancement. Interestingly, the modified sensor shows the significant increase in the sensitivity from 0.7 to 1.14 kPa^{-1} (+62.86%) and the shortening response time from 1.55 to 0.43 s (−72.26%) with respect to the conventional sensor. In addition, the deterioration in elastic behaviour and the faster charge–discharge behaviour observed from the poly(3,4-ethylenedioxythiophene) polystyrene sulphonate mixed polydimethylsiloxane film indicate the better deformation and charge transport than that from polydimethylsiloxane film. Therefore, it can be concluded that the conductive poly(3,4-ethylenedioxythiophene) polystyrene sulfonate

1350911x, 2021, 2, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ell2.12047 by Chung-Ang University, Wiley Online Library on [13/02/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License