Journal paper review on

Outlier Detection by Privacy-Preserving Ensemble Decision Tree Using Homomorphic Encryption

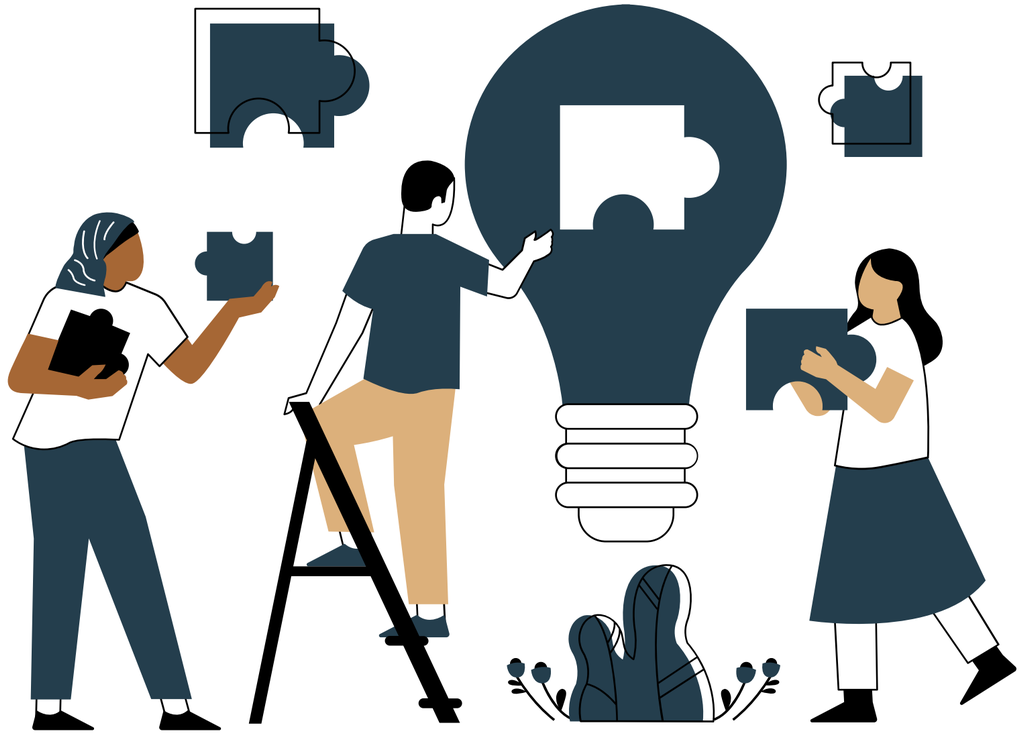
Review - 2

Presented by
Manigandan Ramadasan

Contents

Introduction

- This paper mainly deals with the problem of multiple organizations possess different data sets of a specific task, while they cannot directly share with each other from a privacy point of view.
- A method to **train a Machine Learning model using federated learning** method has been proposed.
- This paper **focus on the outlier detection** under a practical circumstance such that multiple organizations possess different data sets of a specific task.



Preliminaries

Outliers

An outlier is a data point that significantly deviates from other data points in a dataset. It is different from the general trend or distribution of the rest of the data.

Anomaly Detection

Anomaly detection is the identification of rare items, events, or observations which raise suspicions by differing significantly from the majority of the data.

Relation

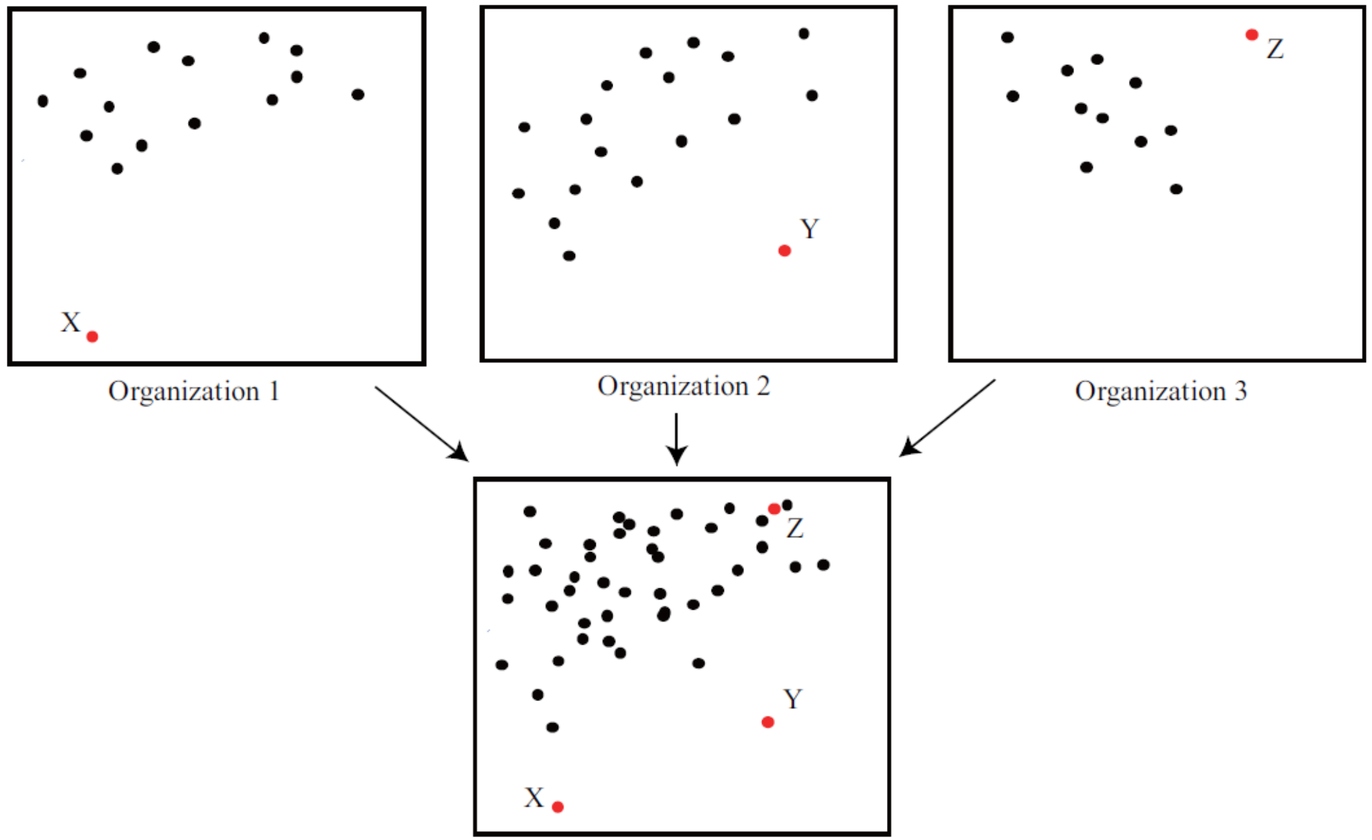
Outliers are anomalous points that are far from other points in a dataset, and anomaly detection is the process of finding these points. So detecting outliers leads to Anomaly detection.

Outlier Detection Algorithm

In standard machine learning algorithms, the goal is often to model the normal behavior of the data. So, there are seperate class of algorithms to detect outliers. Example: Ioslation Forest Algorithm, Local Outlier Factor Algorithm, Histogram Based Approach etc.

Preliminaries (Contd.)

Why Federated Learning using Privacy Preserving Methods?



- Though the red points X, Y, Z seem to be outliers if the three organizations work independently, only X and Y are actually outliers if their data distributions are combined.
- There comes the need of Federated Learning.
- Privacy Concerns arises when confidential data has to be shared for training.
- There comes the needs of Homomorphic Encryption.

Isolation Forest Algorithm

IF is an unsupervised - ensemble learning algorithm that exploits the property that anomalies are in the minority and have attribute values that differ significantly from the majority of the data.

Training Algorithm:

1. When given a dataset, a random sub-sample of the data is selected and assigned to a binary tree.
2. Branching of the tree starts by selecting a random feature first. And then branching is done on a random threshold.
3. This process from second step is continued recursively till each data point is completely isolated or till max depth is reached.
4. The above steps are repeated to construct random binary trees.
5. Then, a forest of such trees, an Isolation Forest, is created by generating multiple Isolation Trees.

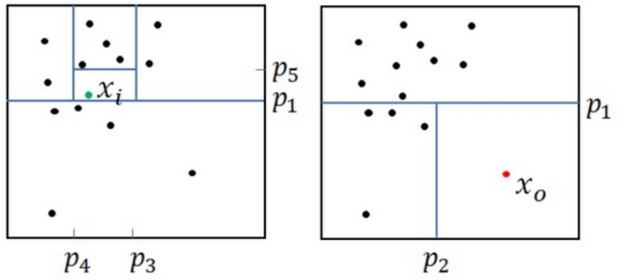
Works effectively for various types of problems with high dimensional and large-scale data.

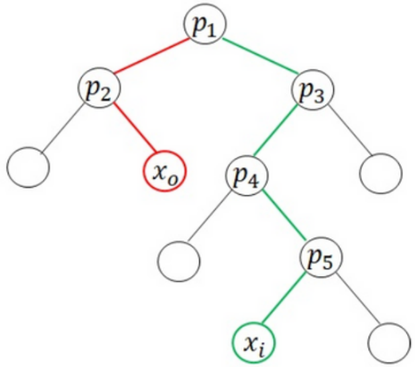
Isolation Forest Algorithm (Contd.)

Decision Making:

The test data is passed through all Isolation Trees, and the depth of the node where a data element is classified is obtained. An anomaly score is calculated based on the average of the depths in all the trees as per the below formula:

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}}$$





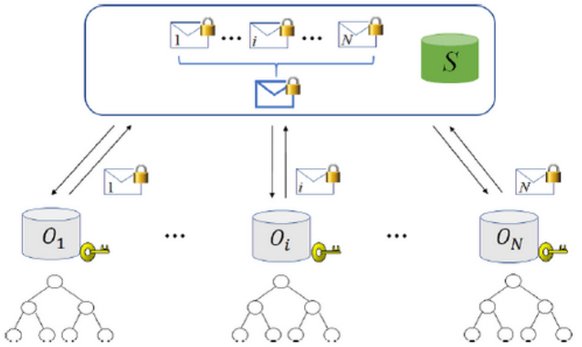
$$c(m) = \begin{cases} 2H(m-1) - \frac{2(m-1)}{n} & for\ m > 2 \\ 1 & for\ m = 2 \\ 0 & otherwise \end{cases}$$

Combining All

Algorithm 1: LocalBuildForest (i, T, ψ, N)

Input : Number of trees N_T, Number of samples ψ, Number of organizations N

1.1 **begin**
1.2 Get a data set D_i of the ith organization.
1.3 Initialize Forest.
1.4 Calculate the maximum depth $l = \log_2 \psi$.
1.5 **for** $j \leftarrow 1$ **to** T **do**
1.6 $X_i \leftarrow \text{sample}(D_i, \frac{\psi}{N})$
1.7 $T_j \leftarrow \text{LocalBuildATree}(X_i, l, 0)$ F$\leftarrow$F $\cup T_j$



- The scheme consists of N organizations namely O1, O2,...., ON.
- ψ is the total number of samples.
- We assume that each organization has ψ/N samples.
- D_i is the dataset of i-th organization
- X_i is the random sub-sampling of the organization's dataset.
- Uses public key additively homomorphic encryption.

Combining All (Contd.)

Algorithm 2: LocalBuildATree (X_i, l, e)

Input : Data in current node X_i^{cur}, Maximum depth l, Current node depth e

Output: A pp-iTree

2.1 **begin**
2.2 if *Tree model completed* **then**
2.3 **return** *pp-iTree*
2.4 $b_i = \text{bool}\{|X_i^{cur}| \geq 2\}$
2.5 Send b_i to server.
2.6 **if** $e = l$ or $split = 0$
2.7 **then**
2.8 Send $\text{Enc}_{pk}(|X_i^{cur}|)$ to server.
2.9 Receive $\text{Enc}_{pk}(|X^{cur}|)$ from server.
2.10 $|X^{cur}| = \text{Dec}_{sk}(\text{Enc}_{pk}(|X^{cur}|))$
2.11 $Size = |X^{cur}|$
2.12 Save $Size$ in pp-iTree.
2.13 **if** $e < l$ **then**
2.14 **if** $split = 1$ **then**
2.15 $q \leftarrow^* [1, \ldots, M]$
2.16 $p \leftarrow^* (\min(X_{i,q}), \max(X_{i,q}))$
2.17 $\text{Enc}_{pk}(q), \text{Enc}_{pk}(p)$
2.18 $q = \text{Dec}_{sk}(\text{Enc}_{pk}(q))$
2.19 $p = \text{Dec}_{sk}(\text{Enc}_{pk}(p))$
2.20 Save p, q in pp-iTree. $X_i^{left} \leftarrow \text{filter}(X_{i,q} < p)$
2.21 $X_i^{right} \leftarrow \text{filter}(X_{i,q} \geq p)$
2.22 LocalBuildATree(X_i^{left}, l, $e + 1$)
2.23 LocalBuildATree(X_i^{right}, l, $e + 1$)

Algorithm 3: ServerBuildATree(N)

Input : Number of organizations N

3.1 **begin**
3.2 Generate the split decision vector $B = [b_1, \ldots, b_N]$.
3.3 **if** $\exists i \in \{N\}, b_i = 1$ **then**
3.4 $L = \{i | i \in \{N\}, b_i = 1\}$
3.5 $r \leftarrow^* L$
3.6 Send O_r $split = 1$.
3.7 Receive $\text{Enc}_{pk}(q), \text{Enc}_{pk}(p)$ from O_r.
3.8 Send $\text{Enc}_{pk}(q), \text{Enc}_{pk}(p)$ to all organizations.
3.9 **else**
3.10 Send $split = 0$ to all organizations.
3.11 $\text{Enc}_{pk}(|X^{cur}|) = \sum_{i=1}^{N} \text{Enc}_{pk}(|X_i^{cur}|)$
3.12 Send $\text{Enc}_{pk}(|X^{cur}|)$ to all organizations.

Combining All (Contd.)

Algorithm 4: OutlierDetection(Forest, Ins, ψ)

Input : trained $pp - iForest$, test data Ins, number of samples ψ, number of trees T

Output: $Score$

4.1 **begin**

4.2 $h(Ins)_{SUM} = 0$

4.3 **for** $T_j \in \mathcal{F}$ **do**

4.4 Classify Ins in T_j

4.5 Get the depth e and the number of data $Size$ for the classified leaf nodes

4.6 $h(Ins)_{SUM} \leftarrow h(Ins)_{SUM} + e + c(Size)$

4.7 However, $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$

4.8 $E(h(Ins)) = \frac{1}{T} h(Ins)_{SUM}$

4.9 $Score = 2^{-\frac{E(h(Ins))}{c(\psi)}}$

- $Size = |X^{cur}|$ is done by the server in an encrypted state, other participants do not know how much data you have in the node.
- Server will know who performed the branch but it wont know any information about the Isolation Forest

Results

- The trained algorithm was tested under two scenarios:
  - Presence of One Organization
  - Presence of Multiple Organizations
- It was also compared with other Outlier Detection Algorithms such as LOF and OCSV.

	iForest	LOF	OCSVM	pp-iForest (N_T)			
				1	2	4	8
Credit Card	0.95	0.78	0.52	0.95	0.95	0.95	0.94
Forest Cover	0.87	0.56	0.66	0.87	0.87	0.87	0.86
Shuttle	0.99	0.56	0.98	1.00	0.99	0.99	0.99
Annthyroid	0.82	0.73	0.57	0.83	0.84	0.85	0.85
Http	1.00	0.35	1.00	1.00	0.99	0.99	0.97
Smtp	0.88	0.30	0.74	0.88	0.87	0.86	0.85

Takeaways

Instead of using traditional ML algorithms, we could use outlier detection algorithms to detect anomalies in our project but the only catch is feature extraction.

We can extend our project by using the scheme proposed in this paper to train the model when datasets from multiple organization is involved.

Ensemble learning algorithms performs better in Anomaly detection when statistical features are given to the model.

State Of Art

- High level overview of the state of art in anomaly detection using encrypted traffic.
- No work is performed on anomaly detection on encrypted traffic payload.
- Only one work is performed on homomorphically encrypted data using statistical features by training Random Forest Algorithm which achieved high accuracy.
- Machine Learning Methods developed:
  - C4.5 - Uses flow data
  - Naive Bayes - Uses packet data
  - Random Forest - Uses flow data
- Deep Learning Methods developed:
  - ANN - Uses Flow data
  - 2D CNN (Architecture similar to LeNet) - Uses Raw Traffic
  - 1D CNN - Uses Raw Traffic
  - Stack and Sparse auto encoder - Network protocol identification method
  - LSTM - Uses Raw Traffic

Conclusion

- Since our main aim is to perform anomaly detection on encrypted traffic payload, it is better to use Deep Learning Models because of textual data.
- CKKS scheme can be used to train the model.
- The scheme proposed in this paper also can be used to train the model when datasets from multiple organization is involved.

Questions and Feedback

Thank You !

References

1. Itokazu, K., Wang, L., & Ozawa, S. (2021, July). Outlier Detection by Privacy-Preserving Ensemble Decision Tree U sing Homomorphic Encryption. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.
2. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In 2008 eighth ieee international conference on data mining (pp. 413-422). IEEE.
3. Akshara. (2023). Anomaly detection using Isolation Forest – A Complete Guide. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/
4. Mavuduru, A. (2022, April 8). How to use Isolation Forests for anomaly detection | Towards Data Science. Medium. https://towardsdatascience.com/how-to-perform-anomaly-detection-with-the-isolation-forest-algorithm-e8c8372520bc
5. Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In 2017 International conference on information networking (ICOIN) (pp. 712-717). IEEE.
6. Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2017, July). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In 2017 IEEE international conference on intelligence and security informatics (ISI) (pp. 43-48). IEEE.
7. Rezaei, S., & Liu, X. (2019). Deep learning for encrypted traffic classification: An overview. IEEE communications magazine, 57(5), 76-81.

References

8. Muliukha, V. A., Laboshin, L. U., Lukashin, A. A., & Nashivochnikov, N. V. (2020, May). Analysis and classification of encrypted network traffic using machine learning. In 2020 XXIII International Conference on Soft Computing and Measurements (SCM) (pp. 194-197). IEEE.
9. Wang, Z. (2015). The applications of deep learning on traffic identification. BlackHat USA, 24(11), 1-10.
10. Gao, N., Gao, L., Gao, Q., & Wang, H. (2014, November). An intrusion detection model based on deep belief networks. In 2014 Second international conference on advanced cloud and big data (pp. 247-252). IEEE.
11. Adiyodi Madhavan, R., & Sajan, A. Z. (2022). Malicious Activity Detection in Encrypted Network Traffic using A Fully Homomorphic Encryption Method.
12. Vu, L., Thuy, H. V., Nguyen, Q. U., Ngoc, T. N., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E. (2018, September). Time series analysis for encrypted traffic classification: A deep learning approach. In 2018 18th International Symposium on Communications and Information Technologies (ISCIT) (pp. 121-126). IEEE.
13. Baldini, G. (2020, June). Analysis of Encrypted Traffic with time-based features and time frequency analysis. In 2020 Global Internet of Things Summit (GIoTS) (pp. 1-5). IEEE.