J. Mantas et al. (Eds.)

doi:10.3233/SHTI200503

Local Distortion Hiding Algorithm in Medical Data: A Case Study Using CART

Georgios FERETZAKIS [a,1], Dimitris KALLES [a] and Vassilios S. VERYKIOS [a]
[a] *School of Science and Technology, Hellenic Open University, Patras 263 35, Greece*

Abstract. Data sharing has become an increasingly common process among health organizations, but any organization will most likely try to hide some sensitive patterns before sharing its data with others. Local Distortion Hiding (LDH), a recently proposed algorithm, has been evaluated only on the assumption of an opponent using the J48 (C4.5) classification algorithm. We now extend the basic approach, and we present a medical dataset hiding case study of a processed by LDH and attacked with the CART algorithm.

Keywords. Privacy-preserving, hiding decision tree rules, medical data privacy

1. Introduction and background

Data privacy is important in health informatics, especially when it comes to analyzing large datasets collected from various sources like health care providers, insurance companies, pharmacies, and research institutions. Information in health care is considered sensitive, and its protection is a major concern when examining the patient's personal health care data for research purposes. Privacy-preserving data mining [1,2] is a research area designed to alleviate issues arising from the use of data mining on data collection, information or knowledge contained in data collections, and the confidentiality of the subjects recorded in them. In our previously published articles [3,4], we suggested a set of techniques to effectively protect against the disclosure of sensitive patterns of information in mining classification rules. We aim to conceal sensitive rules without compromising the information integrity of the entire dataset. The class labels at the tree node corresponding to the tail of the sensitive pattern are changed after an expert selects the sensitive rules to remove the gain achieved by the metric, which causes the split. In the articles [5,6], we expand the above work by formulating a generic look ahead technique that takes into account the structure of the decision tree (DT) from an affected leaf to the root.

LDH (Local Distortion Hiding) algorithm [7] can be used to protect sensitive patterns that result from the use of data mining techniques. In our previous work [8], we presented an application of LDH in an educational dataset and in [9] we applied our technique in a medical dataset that did not, as our previous techniques did, affect the class labels of sensitive instances, but rather modified the values of the attributes of these specific instances. In the proposed approach, we first identify the instances that lead to the development of a particular rule, and then effectively mask this rule with minimal

[1] Corresponding Author, Georgios FERETZAKIS, School of Science and Technology, Hellenic Open University, Patras, Greece; E-mail: georgios.feretzakis@ac.eap.gr.

effect on the rest of the DT by appropriately modifying attribute values. This technique is based on gain ratio, which is the standard metric of C4.5 classifier [10], and it has been tested in different datasets by only using *J48*, the implementation of C4.5 in WEKA [11]. In this paper, we apply LDH algorithm to the same medical dataset as in [9] and test if the under consideration rule is also be hidden by using another classifier than J48, to investigate the resilience of the underlying technique to a variety of attacking opponents.

2. Applying Local Distortion Hashing for Leaf Hiding

In order to hide minimally by modifying the original dataset, we can interpret "minimal" changes in datasets by evaluating if the sanitized DT created by hiding is syntactically identical to the original with minimal modification of the initial dataset. In our example, after the proposed modification, we use the kappa statistic to compare the efficiency of the deduced DT with the original one. The information gain metric is used to select the test attribute at each node of the DT, much like ID3, one of the earliest DT induction algorithms [12], while its successor, C4.5 [10], uses the gain ratio metric, which is an improvement of the information gain. The attribute with the highest gain ratio is chosen as the splitting attribute. Therefore, if we want to suppress a specific attribute test at a node, it would be a reasonable heuristic to seek to change the values of the instances that would enter that node. By this change, the resulting information gain (due to that attribute) will be decreased and being equal to zero where possible. The algorithm LDH locates the parent node of the leaf to be hidden and ensures that the attribute tested at that node will not generate a splitting, which would allow that leaf to re-emerge. CART (Classification And Regression Trees) [13] is a rather common DT classification algorithm based on Gini's impurity index as a splitting criterion.

3. Applying LDH to a medical dataset

We show an example in this section regarding a binary medical dataset from the UC Irvine Machine Learning Repository [14] (SPECT). The SPECT Heart [15] training dataset is based on data from cardiac Single Proton Emission Computed Tomography (SPECT) images. Any patient is grouped into one of two categories, normal or abnormal. SPECT is a good dataset used to check Machine Learning algorithms; it has 187 instances that are described by 23 binary attributes (A1–A22, Class). The binary values for the attributes (A1–A22) are true (t) or false (f), and the values for the Class could be positive (p) or negative (n). We chose for our experiments to use the WEKA [11] framework, an ML software written in Java, and, more specifically, the *J48* and *SimpleCART,* which are the implementations of the C4.5 and CART algorithms, respectively. Both C4.5 and CART algorithms are widely used because of their quick classification and high precision. In our example, we try to hide the terminal node *attr15,* which is shown in the Figures 1 and 2. In this way, we managed to eliminate the node's contribution *attr15* below its parent *attr5*, and the result is shown in Figures 3 and 4. Since the size of the aforementioned DT is too big to fit into a single page, only the section of the DT rules is displayed around the critical node.

```
attr13 = t
|  attr22 = t: p (66.0)
|  attr22 != t
|  |  attr10 = t: p (20.0)
|  |  attr10 != t
|  |  |  attr5 = t
|  |  |  |  attr15 = t: n (1.0)
|  |  |  |  attr15 != t: p (1.0)
|  |  |  attr5 != t
|  |  |  |  attr21 = t
|  |  |  |  |  attr3 = t
|  |  |  |  |  |  attr12 = t: p (2.0)
|  |  |  |  |  |  attr12 != t: n (1.0)
|  |  |  |  |  attr3 != t: p (2.0)
|  |  |  |  attr21 != t: p (11.0)
```

Figure 1. Original DT with the node *attr15*.(J48)

```
attr13!=(f)
| attr22=(f)
| | attr10=(f)
| | | attr5=(t)
| | | | attr15=(t): n(1.0/0.0)
| | | | attr15!=(t): p(1.0/0.0)
| | | attr5!=(t)
| | | | attr21=(t)
| | | | | attr3=(t)
| | | | | | attr12=(f): n(1.0/0.0)
| | | | | | attr12!=(f): p(2.0/0.0)
| | | | | attr3!=(t): p(2.0/0.0)
| | | | attr21!=(t): p(11.0/0.0)
| | attr10!=(f): p(20.0/0.0)
| attr22!=(f): p(66.0/0.0)
```

Figure 2. Original DT with the node *attr15*.(CART)

All dataset files (.arff), before and after hiding has been applied, and the corresponding outputs of both classifiers are available at the website [16].

```
attr13 = t
|  attr22 = t: p (66.0)
|  attr22 != t
|  |  attr10 = t: p (20.0)
|  |  attr10 != t
|  |  |  attr5 = t: p (2.0/1.0)
|  |  |  attr5 != t
|  |  |  |  attr21 = t
|  |  |  |  |  attr3 = t
|  |  |  |  |  |  attr12 = t: p (2.0)
|  |  |  |  |  |  attr12 != t: n (1.0)
|  |  |  |  |  attr3 != t: p (2.0)
|  |  |  |  attr21 != t: p (11.0)
```

Figure 3. Final DT without the node *attr15*. (J48)

```
attr13!=(f)
| attr22=(f)
| | attr10=(f)
| | | attr5=(t): p(1.0/1.0)
| | | attr5!=(t)
| | | | attr21=(t)
| | | | | attr3=(t)
| | | | | | attr12=(f): n(1.0/0.0)
| | | | | | attr12!=(f): p(2.0/0.0)
| | | | | attr3!=(t): p(2.0/0.0)
| | | | attr21!=(t): p(11.0/0.0)
| | attr10!=(f): p(20.0/0.0)
| attr22!=(f): p(66.0/0.0)
```

Figure 4. Final DT without the node *attr15*. (CART)

The kappa statistic adjusts precision by taking into account by chance alone the possibility of a correct prediction.

Table 1. WEKA output for the J48 and CART classifiers on original and modified datasets.

	Original		**Modified**	
	J48	**CART**	**J48**	**CART**
Correctly Classified Instances	182	182	181	181
Incorrectly Classified Instances	5	5	6	6
Kappa statistic	0.834	0.834	0.795	0.795
Mean absolute error	0.0352	0.0326	0.0406	0.0379
Root-mean-squared error	0.1328	0.1276	0.1425	0.1377
Relative absolute error	29.296%	21.529%	26.831%	25.064%
Root relative squared error	48.866%	46.976%	52.442%	50.686%

The kappa statistic values corresponding to both classifiers (J48, CART) on original and modified datasets are exactly the same, 0.834 and 0.795, respectively. From all other WEKA statistics presented in Table 1, we can also conclude that the node *attr15* was successfully hidden without significantly affecting the effectiveness of the resulting DT.

The under investigation part of the DT induced by the modified dataset produced by LDH algorithm is the same as the original one without the node *attr15* for both classifiers.

4. Conclusions

Our methodology (LDH algorithm) enables one to determine which DT leaves should be hidden in the original dataset, and then to adjust those attribute values in specific instances. Even though LDH algorithm is based on gain ratio metric - the one that is used by C4.5 – in this work, we presented a case study by testing the resulting dataset in another classifier (CART), which is based on another metric (Gini). We observed that the under consideration decision rule in this medical dataset was successfully hidden on both classifiers without compromising the information value in rest classification rules. We expect LDH to demonstrate similar resilience to any DT induction algorithm, which uses information-based splitting criteria, and we plan to confirm this expectation by experimentation.

References

[1] Verykios VS, Bertino E, Fovino I, Provenza L, Saygin Y, Theodoridis Y, State-of-the-art in privacy-preserving data mining, ACM SIGMOD 33 (2004), 50.

[2] Brumen B, Heričko M, Sevčnikar A, Završnik J, Hölbl M, Outsourcing Medical Data Analyses: Can Technology Overcome Legal, Privacy, and Confidentiality Issues?, J. Med. Int. Res. 15 (2013), 283.

[3] Kalles D, Verykios VS, Feretzakis G, Papagelis A, Data set operations to hide decision tree rules, In Proceedings of the Twenty-second European Conference on Artificial Intelligence, Hague, The Netherlands, 29 August–2 September 2016.

[4] Kalles D, Verykios VS, Feretzakis G, Papagelis A, Data set operations to hide decision tree rules, In Proceedings of the 1St International Workshop on AI for Privacy and Security, Praise '16, Hague, The Netherlands, 29–30 August 2016.

[5] Feretzakis G, Kalles D, Verykios VS, On Using Linear Diophantine Equations for Efficient Hiding of Decision Tree Rules, In Proceedings of the 10th Hellenic Conference on Artificial Intelligence—SETN '18, Patras, Greece, 9–12 July 2018.

[6] Feretzakis G, Kalles D, Verykios VS, On Using Linear Diophantine Equations for in-Parallel Hiding of Decision Tree Rules, Entropy 21(1) (2019), 66.

[7] Feretzakis G, Kalles D, Verykios VS, Using Minimum Local Distortion to Hide Decision Tree Rules, Entropy 21(4) (2019), 334.

[8] Feretzakis G, Kalles D, Verykios VS, Local Distortion Hiding in Financial Technology application: a case study with a benchmark data set, 10th International Conference on Information, Intelligence, Systems and Applications (IISA) (2019).

[9] Feretzakis G, Kalles D, Verykios VS, Hiding Decision Tree Rules in Medical Data: A Case Study, Studies in Health Technology and Informatics, Health Informatics Vision: From Data via Information to Knowledge 262 (2019), 368 – 371.

[10] Quinlan JR, C4.5. Programs for Machine Learning. Morgan Kaufmann: CA, USA, 1993.

[11] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I, The WEKA data mining software. ACM SIGKDD Explor. Newslett. 11 (2009), 10–18.

[12] J. R. Quinlan, Induction of Decision Trees. In Machine Learning 1, Kluwer Academic Publishers: Boston, MA, USA, (1986), 81–106.

[13] Breiman L, Classification and regression trees, The Wadsworth and Brooks-Cole statistics-probability series, Chapman & Hall, 1984.

[14] Dua D, Graff C, (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[15] Kurgan LA, Cios KJ, Tadeusiewicz R, Ogiela M, Goodenday LS, Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis. Artif. Intell. Med. 23 (2001), 149–169.

[16] Feretzakis G, LDH Algorithm in Medical Data: A Case Study using CART, Available online: http://www.learningalgorithm.eu/datafiles_ICIMTH2020.html (accessed on 22 March 2020).