ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Ensuring Privacy in Data Mining using Neural Networks.

D Niranjan, G Manikandan*, N Sairam, V Harish, and Nooka Saikumar.

School of Computing, SASTRA University, Thanjavur, Tamilnadu, India.

ABSTRACT

Data mining is used to extract valuable details and patterns from the large repositories. The key area of apprehension is that non-sensitive data could convey insightful information such as facts and interesting patterns. Privacy preserving data mining (PPDM) is a new track in data mining which focuses on hiding individual's characteristics without compromising the data usability. The primary idea of privacy preserving data mining is to perform data mining on confidential data. In the proposed approach we have formulated rules for the given data set. A Neural Network is constructed based on these rules. The output of the neural network is used as a noise for perturbing the original data. Experiments were conducted using the dataset available in the UCI machine repository. K-Means clustering algorithm is used for the validation purpose. This algorithm is applied to the original data (D) and the modified data (D') to compute the misclassification error.

Keywords: Data Privacy, Data anonymity, Data Utility, Big Data, Neural Network

Corresponding author

INTRODUCTION

Data privacy is an approach that prevents information from unauthorized users. The Data owners do not disclose the sensitive data to unauthenticated persons for any cost. Data privacy is different form data security, Data privacy is building rules or polices to ensure the sensitive data being preserved [1]. Security is protecting the data from being stolen. Degree of utilization for privacy preserved data is more than security enabled data. Increasing privacy and data utilization is a challenging task [2].

Artificial Neural Network is a learning system based on machine learning and cognitive science. It is a network model, where the processes of data are taken as network nodes. Each neural network can be categorized into three levels. First layer is for input and second one is the Hidden layer. The data from input layer will be sent to hidden layer [3]. In Hidden layer the data is processed based on output. Input layer sends values along with its weightage value to the hidden layer. In every step, the weightage value will be changed until the hidden layer is reached and at last the processed output will be sent to output layer [4].

There are so many techniques and approaches like Randomization method, k-anonymity model and l-diversity, distributed privacy preservation, Downgrading Application Effectiveness implemented for privacy preservation and Data Utilization [5-7]. But these techniques lack in data utilization in privacy preserved data. In this paper we provide an approach for privacy preserving and also for effective data utilization. This paper also explains how to add noise that provides effective data utilization.

Neural Network

Neuron: The cell that performs information processing in the brain. Neural network is an Information Processing Model based on biological nervous system like brain process information. It is composed of inter-connected processing neurons. Neural network can be called a student that can learn from examples. The advantage of neuron is that it works in parallel manner. It can acquire, store, and utilize learning experience.

Neural Network – Architecture

Neural Network is implemented based on biological nervous system. The following figure represents biological neuron.

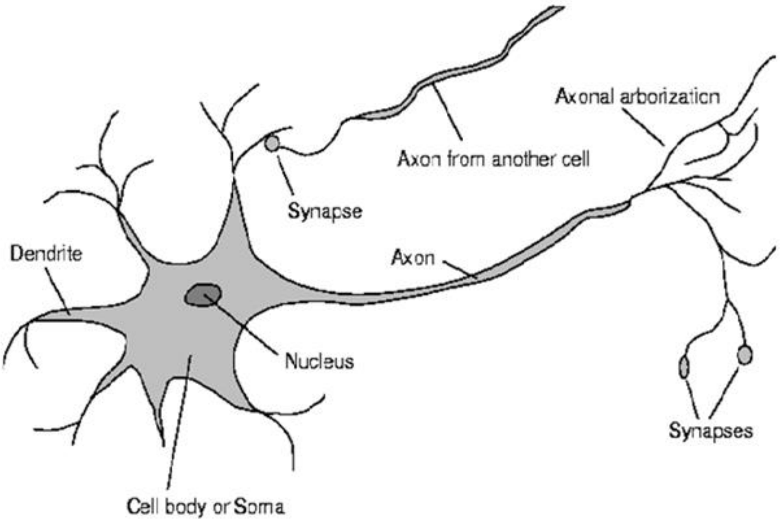


Figure 1: Biological Neuron

In biological neuron system, the neuron system in brain transmits the signal from one neuron to another neuron. The information will be determined whether the neuron send signal to another or not. Signal sending depends on the strength of the bond (synapse) between two neurons.

Neural network: Computing Elements

The neuron which is present in Artificial Neural Network (ANN) comprises of input, processing, and output. The input neuron receives input from other neuron or environment. The processing neuron, in other words, the hidden neurons process the received information. The processed information will be sent to output neuron, the output neuron sends the output to other neuron or is sent back as input to fine-tune the result [8-9]. A neural network has an initial weight. The weight can either be randomized or can be a manual weight. Neuron learning plays a vital role, so there is a need for a learning algorithm which may be supervised or unsupervised. Irrespective of the learning algorithm chosen, we need to give training examples for learning the neurons and assign examples as input to output neurons.

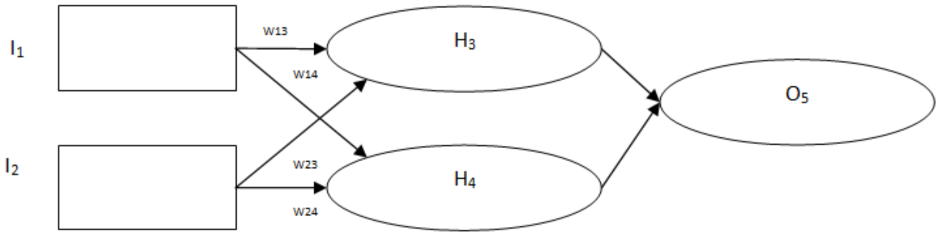


Figure 2: Feed-Forward Network with two inputs, two hidden nodes, and one output node

PROPOSED SYSTEM

One of the major tasks is to increase the data utilization in privacy preserved data. So, we have proposed perturbation algorithm**.** Perturbation algorithm is used to preserve privacy and utilize the perturbed data. This algorithm works by adding random noise in each domain value where the original value will be replaced into modified data. But this approach ensures data privacy, whereas the utilization of data will not be reduced or suppressed.

Overall Flow Diagram

The overall flow diagram of proposed system is graphically represented here.

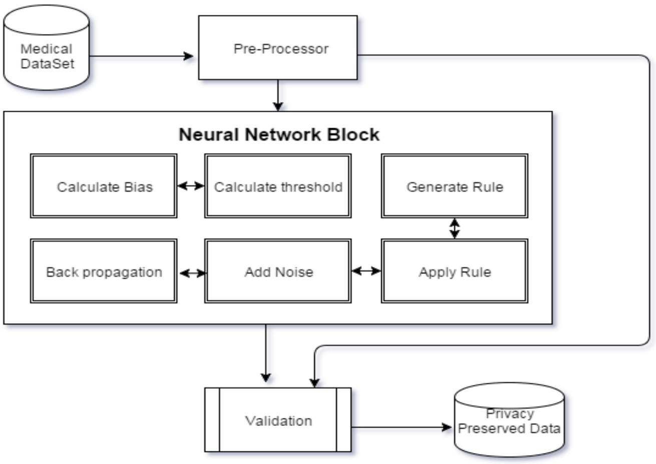


Figure 3: Flow of data from input to output

The diagram shows the flow of data from input dataset to final output. In the proposed approach we use medical dataset; the data needs to be pre-processed for removing noisy data and missing values. The pre-process is done by using “Weka tool”. After pre-processing, the data is sent to neural network block. In Neural network block we initially find the co-relation between data values. The value obtained as co-relation is used as noise. We generate the rules based on data value, and add the noise to each data item based on the generated rules. After the noise addition, the noise added modified data and original data are compared for calculating accuracy and performance. The validation part is done by using K-Means clustering algorithm.

Generating Noise

In this approach we add noise to each and every domain values. The noise added data should not reduce data utilization at the same time it should preserve privacy. The main thing is the data consistency must be maintained. For example, let's take example of medical dataset, the patient record in the 7th row shows that he has diabetes. Even after the addition of noise, the privacy of patient should to be preserved. So to generate meaningful noise here we generate co-relation between data. Co-relation is a measurement of statistical dependencies or relationship between data.

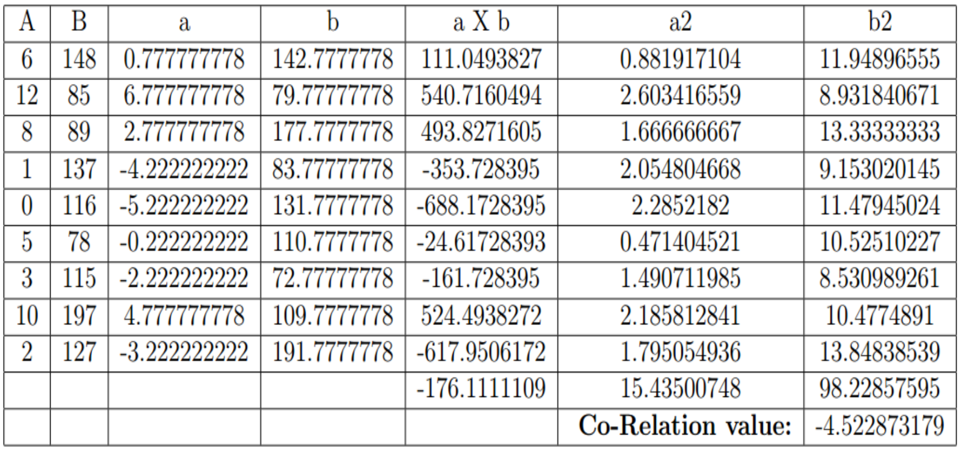
Generating Rules

Rules are the condition which is used to filter the domain value from dataset. Based on rules we add the amount of noise in domain value. Noise is generated based on co-relation between data. For each data item we apply rules and add the appropriate noise to data item.

Co-Relation between data

Here we use Population Pearson's Co-Relation technique

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - x')(y_i - y')}{\sqrt{\sum_{i=1}^{n}(x_i - x')^2 \sum_{i=1}^{n}(y_i - y')^2}}$$



A	B	a	b	a X b	a2	b2
6	148	0.777777778	142.7777778	111.0493827	0.881917104	11.94896555
12	85	6.777777778	79.77777778	540.7160494	2.603416559	8.931840671
8	89	2.777777778	177.7777778	493.8271605	1.666666667	13.33333333
1	137	-4.222222222	83.77777778	-353.728395	2.054804668	9.153020145
0	116	-5.222222222	131.7777778	-688.1728395	2.2852182	11.47945024
5	78	-0.222222222	110.7777778	-24.61728393	0.471404521	10.52510227
3	115	-2.222222222	72.77777778	-161.728395	1.490711985	8.530989261
10	197	4.777777778	109.7777778	524.4938272	2.185812841	10.4774891
2	127	-3.222222222	191.7777778	-617.9506172	1.795054936	13.84838539
				-176.1111109	15.43500748	98.22857595
					Co-Relation value:	-4.522873179

Table 1: Co-Relation

From the above table we clearly get -**4.522873179** as co-relation value. This value will be added to original data as noise. Then the resultant value will be send to back propagation for get good results. Each iteration has separate error value, the amount of error reduced is the amount of accuracy we achieve.

Back propagate the output

Back propagation works under supervised learning method. To reach the maximum amount of data utilization and privacy preservation, we use back propagation algorithm (i.e.) the output of neural network block is given as input. It may take several iterations to fine tune its results. The process cycle stops when the error value is reduced. To reduce noise the weight of input keeps on updating its value.

EXPERIMENTAL RESULTS

The implementation is done by using java language and Neuroph, a distributed and open-source neural network framework. The Noise added domain values play their role efficiently in securing data privacy and data utilization. We use medical diabetes dataset for input. Dataset contains 1000 patient records and 4 arguments, for accuracy purpose we take only two arguments. In this section we discuss the experimental analysis result by using graphical form. . The figure 4 shows the raw data and the figure 5 represents the noise added data Result shows that the amount of data anonymous by using this proposed approach

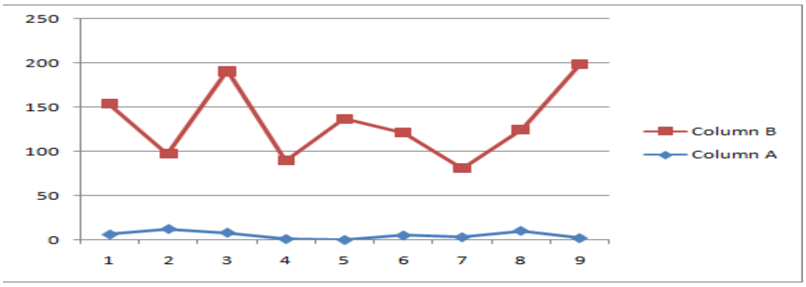


Figure 4: Graph Representing Raw Data

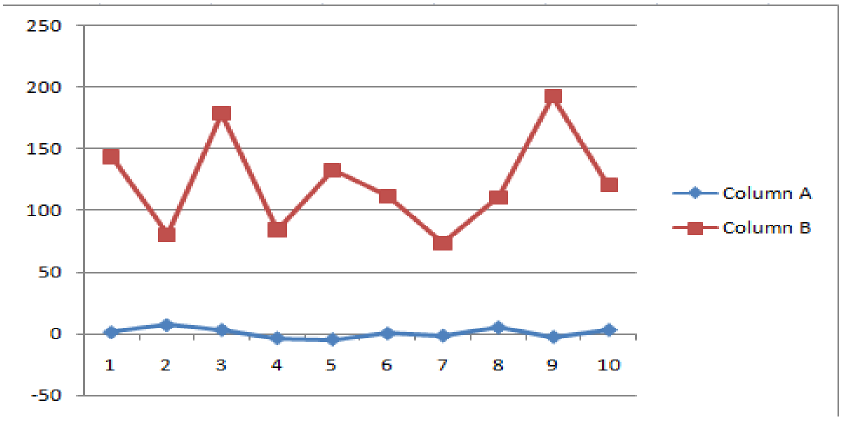


Figure 5: Graph Representing Anonymous Data

CONCLUSION

In each and every encryption approach, the user cannot easily read the data until the data is decrypted using the decryption key. The impact will be more if the hacker finds both the data and its appropriate key.The techniques like perturbation can be used as one of the solutions to solve this problem. In this paper, we have added a meaningful noise to a sensitive attribute, so that we can preserve data privacy as well as the data utilization.

REFERENCES

[1] Unil Yun , Jiwon Kim. Expert Systems with Applications 2015; 42: 1149 – 1165.
[2] Nissim Matatov , Lior Rokach , Oded Maimon. Information Sciences 2010; 180:2696 – 2720.
[3] Jun-Lin Lin ,Yung-Wei Cheng. Expert Systems with Applications 2009; 36: 5711 – 5717.
[4] Hemanta Kumar Bhuyan , Narendra Kumar Kamila. Applied Soft Computing 2015; 36: 552 – 569.

[5] Erez Shmueli, Tamir Tassa. Information Sciences 2015; 298 : 344 – 372.
[6] Jieh-Shan Yeh , Po-Chiang Hsu. Expert Systems with Applications 2010; 37: 4779 – 4786.
[7] Josep Domingo-Ferrer , Jordi Soria-Comas. Knowledge-Based Systems 2015; 74: 151 – 158.
[8] M. Prakash, G. Singaravel. Computers & Electrical Engineering 2015; 45: 134 – 140.
[9] Florian Kohlmayer, Fabian Prasser, Claudia Eckert, Klaus A. Kuhn. J Biomed Inform 2014; 50: 62 – 76.