Improving Data Privacy Using Fuzzy Logic and Autoencoder Neural Network

Sayantica Pattanayak
Department of Computer Science
North Dakota State University
Fargo, ND, USA
sayantica.pattanayak@ndsu.edu

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND, USA
simone.ludwig@ndsu.edu

***Abstract*—Data privacy is a very important problem to address while sharing data among multiple organizations and has become very crucial in the health sectors since multiple organizations such as hospitals are storing data of patients in the form of Electronic Health Records. Stored data is used with other organizations or research analysts to improve the health care of patients. However, the data records contain sensitive information such as age, sex, and date of birth of the patients. Revealing sensitive data can cause a privacy breach of the individuals. This has triggered research that has led to many different privacy preserving techniques being introduced. Thus, we designed a technique that not only encrypts / hides the sensitive information but also sends the data to different organizations securely. To encrypt sensitive data we use different fuzzy logic membership functions. We then use an autoencoder neural network to send the modified data. The output data of the autoencoder can then be used by different organizations for research analysis.**

***Keywords*-privacy, security, fuzzy logic, autoencoder**

I. Introduction

The latest advancements in the field of computer science have encouraged many organizations to store data about individuals for efficient decision making. To make the decision making more efficient and precise, there is a need to collaborate with other organizations. These different organizations analyze the raw data. However, analyzing raw data can cause threats to privacy [1].

Although everyone understands the concept of privacy, there is no universally accepted definition of privacy. Privacy can be defined in three ways. *Privacy in Information* deals with the handling and collection of private data. *Communication Privacy* deals with privacy while communicating. *Territorial Privacy* is concerned with the invasion of physical boundaries. Our work focuses on privacy in information. Information about criminals, patients, and financial transactions are sensitive.

When the raw data is shared there is a chance of a privacy breach. For instance, banks might wish to collaborate in order to detect the fraudulent behavior of customers. This requires the bank to share financial records of the customer. Also, hospitals want to share data of patients with the other hospitals for efficient diagnosis of diseases. In both cases, the bank and the hospitals hold shared data without violating the privacy of the individual customer. Therefore, a technique is required to share data while preserving individuals' privacy. The technique should hide the attributes in the data set which identify the individual. Also, the shared data set should be equivalent to the raw data set. Then, we can assure any individual not to be scared of sharing their sensitive information. Along with that we can assure other organizations that the shared data is the same as the modified data.

Therefore, many approaches have been employed for privacy preserving. The approaches which are used were randomization [2], anomization [3], secure multiparty computation [4]. Also, data perturbation methods have also been used to add noise in the raw data. These approaches were used to hide the original data. Then, to share the data efficiently data mining techniques were used for these approaches. Data mining approaches have shown that the original and the modified data are relatively similar. The accuracy of these approaches are measured using different classifiers.

Our paper focuses mainly on data stored in hospitals. The hospitals store the sensitive information of patients in the form of electronic health records (EHR). This data is sent to different other hospitals or data analysts for further research. This paper introduces a new technique to send data to different organizations while hiding the sensitive attributes of the patients. Our new technique uses fuzzy logic and artificial neural network (ANN) to share the data among different organizations.

The remainder of the paper is organized as follows. Section II contains the literature survey. Section III provides the background of fuzzy logic and autoencoder neural network. Section IV describes our proposed approach. Section V includes experiments and evaluation. Section VI presents and discusses the results. Section VII summarizes the findings.

II. Related Work

There has been a surge in recent research activity in privacy preserving of sensitive data. Several papers have been published on various aspects of privacy preserving. We discuss some of the previous work related to our approach.

Several data hiding techniques exist based on different assumptions. Data swapping is the commonly used data hiding technique. It refers to a method of swapping information from one record to another [5], [6]. The amount of swapping to be done in a database is dependent on the application and

the need of the organization. There exist various variants of swapping. The records for swapping are purposely or randomly chosen since they are believed to have a greater risk of re-identification. The advantage of swapping is that it can be easily implemented and is one of the best methods of preserving confidentiality. Its main disadvantage is that even with a very low swapping rate, it can destroy analytic properties, particularly sub-domains.

To overcome the disadvantage of the earlier two methods, the authors introduced a controlled way of swapping in 1995 [7]. In this approach, the values of an individual record are sorted and swapped in a range of k-percent of the total range. Randomization determines the specific values of the record value to be swapped. The procedure is repeated for each variable until all variables have been swapped. The main disadvantage of this approach is the determination of k. If k is relatively small, then analytic distortions on the entire file may be small for simple regression analysis. If k is large, there is an assumption that the re-identification risk may be reduced [8]. The authors provided methods for aggregating several attributes simultaneously. The methods are based on multivariable metrics for clustering variables into the most similar groups. The methods are not as easily implemented because they can involve sophisticated optimization algorithms. For computational efficiency, the methods are applied from two to four attributes simultaneously whereas many public use files contain 12 or more attributes. The advantage of the multivariable aggregation method is that it provides better protection against re-identification. Its disadvantage is that analytic properties can be severely compromised, particularly if two or three uncorrelated attributes are used in the aggregation process. The attributes that are not micro-aggregated may themselves allow re-identification.

Micro-aggregation is another technique for data masking [10], [9]. It aggregates the record values of attributes and is intended to reduce the re-identification risk. In single ranking micro-aggregation, each attribute is aggregated independently of other attributes. The method is easy to implement and the values of attributes are sorted and divided into groups of size k. In practice, k is taken to be three or four to reduce analytic distortions. In each group, the values are replaced by an aggregate such as the mean or the median. The micro-aggregation is repeated for each of the attributes that are considered to be usable for re-identification.

Adding noise to the data sets showed that it is theoretically possible to recover the mean and covariance of a given record for arbitrary sub-domains [11], [12]. Both of the papers showed that the masked data set, by adding noise, provides good analytic properties such as regression analysis that closely reproduces regression analysis of the unmasked data. The authors reasoned that adding noise can yield files with moderate re-identification rates. In 2017, the authors in [13] used symmetric key encryption to hide the data before sending it to the other party using neural network. In 2018, the same authors [14] used the hiding technique to encrypt the password to log into multiple servers. However, both encryption approaches used the symmetric key method. In symmetric key encryption the shared secret key has to be secured. In 2017, the authors in [15] came up with the idea of adding noise to the data set using neural networks. The authors achieved privacy by hiding two of the attributes. The disadvantages of using this approach is that they classified age into groups, i.e., two different ages will fall into the same group.

In [16], the authors introduced fuzzy logic for privacy preserving. The authors claim their technique is useful for both numerical and categorical attributes. However, the authors did not use any data mining technique to prove that their modified data is the same as the raw data. The authors in [17] published a paper showing a comparative study of data perturbation. They also used fuzzy logic to preserve privacy. The authors used different classifiers like SVM, ID3 and C4.5 on the original as well as on the perturbed data.

Although different techniques have been applied to preserve privacy, however, each of them has their own limitations. First of all, the data swapping method has a very low swapping rate. Then, in order to overcome the randomization problem, micro-aggregation was used. Both of these techniques have a limitation in determining the split of the data records. In order to avoid this limitation, a data set was created by adding noise using fuzzy logic and neural networks. The authors of the papers however did not show how to send the data securely to different other organizations. In addition, the authors added noise only to the attributes that are numerical and categorical.

In our paper, we used fuzzy membership functions and an autoencoder neural network to modify the sensitive information. Our proposed technique not only hides numerical and categorical attributes but also real valued attributes. After the fuzzification of the three sensitive attributes, we send the modified data to an autoencoder neural network. The current approach retains both privacy and the accuracy of the result. Our paper is different from the earlier work done so far by combining fuzzy logic with an autoencoder neural network.

III. Background

This section explains the background information used in our proposed approach. In our proposed approach we used fuzzy membership functions [18] and an autoencoder neural network.

A. Fuzzy Logic

Fuzzy logic was introduced in 1965 by Zadeh in his research paper "Fuzzy Sets" [19]. He is considered as the father of fuzzy logic. Fuzzy logic resembles the human decision-making methodology by dealing with vague and imprecise information. Fuzzy logic is applied to many real-world problems since it is based on degrees of truth rather than based on Boolean logic. Fuzzy logic is best understood within the context of set membership. Basically fuzzy logic allows partial membership, which means that it contains elements that have varying degrees of membership within the set. Furthermore, membership functions characterize fuzziness whether the elements in the

fuzzy sets are discrete or continuous. We have used different membership functions in our approach, which are explained below.

B. Mathematical Notation

Here, we elaborate on the different membership functions which have been used in our paper to perturb the raw data. Figure 1 shows the plot of different membership functions.

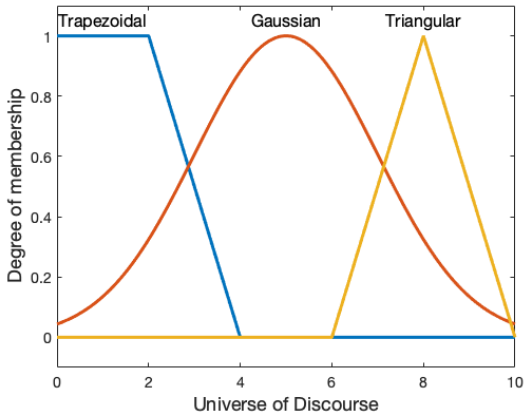

Fig. 1. Different Membership Functions

1) Triangle Membership Function: This membership function uses the following relationship:
$TriMF(D:X,Y,Z) = 0$ when $D < X$
$= (D-X)/(Y-D)$ when $X \leq D \leq Y$
$= (Z-D)/(Z-Y)$ when $Y \leq D \leq Z$
$= 0$ when $D > Z$
Here, D represents the value in the data set. X, Y and Z are three boundary points.
2) Gaussian Membership Function: This membership function has the following relation:
$GaussMF(D:C,W) = EXP{-}(D-C)^2/W^2$
In the above relation, D is the value in the data set. C is the center, and W is the width of the function.
3) S-Shaped Membership Function: For this membership function we have the following relation:
$SMF(D:X,Y) = 0$ when $D \leq X$
$= 2*[(D-X)/(Y-X)]^2$ when $X \leq D \leq (X+Y)/2$
$= 1{-}2{*}[(D{-}Y)/(Y{-}X)]^2$ when $(X{+}Y)/2 \leq D \leq Y$
$= 1$ when $D \geq Y$
where D is the value in the data set, X is the minimum, and Y is the maximum value in the data set.

Our model is based on privacy and security. Hence, we kept the boundary points of the triangular membership functions, the center and the width parameters of the Gaussian Membership functions and the maximum and minimum values of the s-shaped membership functions secured. These values are only known to the sender and the receiver.

C. Artificial Neural Network

Artificial neural networks are one of the main tools used in machine learning. As the "neural" part of the name suggests, the networks are brain-inspired systems, which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. Artificial neural networks are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract or to teach a machine to recognize. While neural networks have been around since the 1940s, it is only in the last several decades that they have become a major part of artificial intelligence.

1) Autoencoder: In our work we used an autoencoder [20] neural network. The autoencoder has a multilayer percepepton (MLP) like structure with input layer, hidden layer and output layer. However, unlike an MLP, autoencoder neural networks do not require any target data since the network tries to learn the input itself. The autoencoder consists of two parts: encoder and decoder. The encoder compresses the inputs to the most important features. The decoder reconstructs the original input from the encoder. The hidden layer compresses the input of the most important feature vectors. Therefore, the number of feature vectors are reduced in the hidden layer. Autoencoders are very frequently used for dimensionality reduction and feature selections.

D. Proposed Model

Our proposed model is divided into two tasks. The first task is based on hiding the sensitive information. The second task is to send the perturbed data to different organization using an autoencoder. Thus, in our proposed approach in order to accomplished the first task we used fuzzy membership functions to hide the sensitive attributes. These are the attributes which the patient does not want to share. By hiding the data using different fuzzy membership functions this will make it difficult for some other party to identify the patient. Next, to send data to different organizations we used an autoencoder. By using an autoencoder we can keep the raw data and the perturbed data set almost similar otherwise no other organization would be interested in a completely perturbed data set.

IV. Our Approach

In our paper, we focused on how to improve the data privacy of the patients. Data privacy could be improved if we could hide the sensitive information of the patient before sending the data to different organizations. Also, the modified data and the raw data should almost be the same. Otherwise the modified data set will be misleading. The research analyst will not be interested in using such a modified data set.

In our approach, we selected a recent cervical cancer (risk factor) data set from the UCI data repository [21]. This data set focuses on the prediction of indicators / diagnosis of cervical cancer. The data set was collected by the "Hospital Universitario de Caracas" in Caracas, Venezuela. The data set compromises demographic information, habits and historic medical records of 858 patients. The 36 attributes are boolean

or real valued. The data set has 4 target variables, which are Hinselmann, Schiller, Cytology, Biopsy.

We first formatted the data set since an autoencoder learns the feature vector and reconstructs the output. We included the target variables in the feature set. We then sent all the features as the feature vector to an autoencoder. We modified the data set using the following modifications:

1) First, we identified the sensitive attributes from the data set. We identified age, number of sexual partners, and first sexual intercourse as sensitive information. This is the information which most of the patients do not want to share since their identity could be revealed by sharing this information.
2) We perturbed or hid these three attributes using three fuzzy membership functions. The three fuzzy membership functions are S-shaped, Gaussian and Triangular. These three membership functions are described in the background section. These membership functions require boundary points, which are selected by the organization which owns the data set. In our approach, the hospital which has the raw data will select the boundary points. These boundary points should be securely stored by the hospital. The boundary points should not be shared with any other organization since these could reveal the patients' identity. The rest of the attributes are left as they are in the raw data set.

To share the data with different organizations we used an autoencoder neural network. We are not interested in the class which predicts the cervical cancer rather we are interested in the feature vectors as the output. Thus, we consider all the features including the target variables as feature vectors. An autoencoder is one of the neural networks which learns its own input. Thus, we used an autoencoder to send the data set to different other organizations.

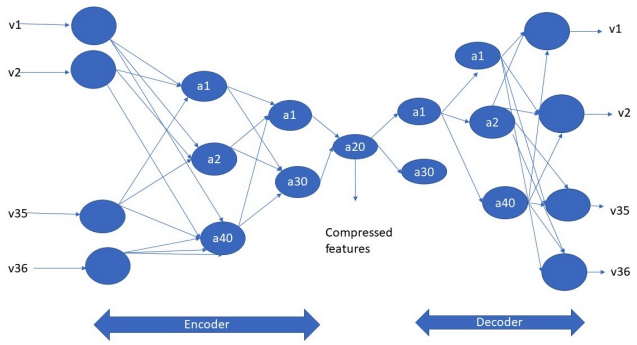

Fig. 2. Autoencoder Neural Network

Figure 2 outlines the autoencoder used in our approach. From the figure we can see that we used 36 feature vectors as input. The hidden layer compresses the 36 feature vectors to 20. These 20 feature vectors are the most important attributes. The decoder reconstructs those 36 attributes from the encoder. Since the autoencoder neural network is reconstructing the 36 feature vectors again, we measure the accuracy with different loss functions. The output of the autoencoder is sent as a data set to different organizations. The data set shares the same information as the raw data set without compromising the privacy of the patients.

V. Experimental Setup

The cervical cancer data set which we used in our experiment consists of 36 attributes. Among them we consider the first three attributes as sensitive. We have hidden those attributes with three fuzzy membership functions. The three fuzzy membership functions which we used are Gaussian, S-shaped and Triangular. Thus, in total we have 42 attributes instead of 36 attributes in our data set.

Next, we have to send the modified data set to another party using an autoencoder. We dropped two columns 'Time since first diagnosis' and 'Time since last diagnosis' since these two columns have the most missing values. Thus, we are left with 40 columns. Although we have 40 attributes, but at one time we are sending only 34 attributes through the autoencoder. These 34 attributes exclude the attributes created by different membership functions. These 34 attributes are the feature vectors/input to the autoencoder. The hidden layer/layers tries to compress the input features to a latent space representation. The output is reconstructed from this representation. This is how the output of our autoencoder is created. The reconstructed output includes all feature vectors. To experiment with different data sets, the following three have been generated based on:

1) Dropping the columns which has missing values.
2) Dropping the rows which have missing values.
3) Replacing the missing values with the mean of that particular column.

VI. Evaluation and Results

In this section, we summarize our observations and results. We measure the accuracy against different loss functions and also with and without sparsity constraints enabled in the autoencoder. The sparsity constraints regularize the autoencoder [22].

The parameters which are considered are as follows:

- Random State = 150
- Epoch = 500
- Optimizer = Adadelta
- Activation function = tanh
- train split= 0.8
- activity regularizers= 0.000001

Table I shows the accuracy of different membership functions. We considered the modified data set without sparsity constraints by dropping the missing columns. From the table we can see that Logcosh used as the loss function achieves the best value of 81.60% for the three different membership functions. Also, we can say from the values in the table that MSE (Mean Square Error) as a loss function obtains the best accuracy of 81.15% for the S-shaped function.

In Table II, we evaluated the same data set with the sparsity constraints. We can see that the Gaussian membership function

TABLE I
ACCURACY AND LOSS VALUE OF DROP COLUMN DATA SET AND WITHOUT SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	0.1313	0.7588
Gaussian	MSE	0.1278	0.8020
Gaussian	Categorical_crossentropy	1.0206	0.0207
Gaussian	Logcosh	0.1364	**0.8160**
Gaussian	Hinge	0.7107	0.2447
S-shaped	Mean_Absolute	0.1316	0.7835
S-shaped	MSE	0.1272	**0.8115**
S-shaped	Categorical_crossentropy	1.0756	0.0166
S-shaped	Logcosh	0.1280	**0.8127**
S-shaped	Hinge	0.7038	0.2606
Traingular	Mean_Absolute	0.1356	0.7769
Triangular	MSE	0.1273	0.8063
Triangular	Categorical_crossentropy	0.9532	0.0433
Triangular	Logcosh	0.1280	**0.8146**
Triangular	Hinge	0.6738	0.2142

with MSE as the loss function obtains the best accuracy of 87.34%.

TABLE II
ACCURACY AND LOSS VALUE OF DROP COLUMN DATA SET AND WITH SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	0.1313	0.7409
Gaussian	MSE	0.1260	**0.8734**
Gaussian	Categorical_crossentropy	0.9451	0.0058
Gaussian	Logcosh	0.1263	0.8122
Gaussian	Hinge	0.7091	0.1135
S-shaped	Mean_Absolute	0.1326	0.8180
S-shaped	MSE	0.1261	0.8457
S-shaped	Categorical_crossentropy	1.0220	0.0291
S-shaped	Logcosh	0.1266	0.8239
S-shaped	Hinge	0.6954	0.1397
Triangular	Mean_Absolute	0.1313	0.8122
Triangular	MSE	0.1278	0.8457
Triangular	Categorical_crossentropy	0.0936	0.0015
Triangular	Logcosh	0.1267	0.8384
Triangular	Hinge	0.7059	0.1266

Then, we evaluated the data set by dropping the rows which have missing values and ran the autoencoder without the sparsity constraints. The results are given in Table III. From the table we can summarize that the Gaussian and Triangular membership functions with hinge value both obtains the best accuracy of 62.51%. Table IV summarizes the evaluation of the data set running the autoencoder with the sparsity constraints. We can see that the Gaussian membership function with logcosh obtains the best accuracy of 62.51%.

We also evaluated the data set in which the missing values are replaced by the mean of that particular column. We evaluated that data set running the autoencoder without the sparsity constraints. The results are given in Table V. We can summarize that the Gaussian membership function with logcosh as the loss function returns the best value of 62.52%. From Table VI we can say that the Gaussian membership function with MSE and using the sparsity constraints returns

TABLE III
ACCURACY AND LOSS VALUE OF DROP ROW DATA SET WITHOUT SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	4.4915	0.1458
Gaussian	MSE	4.2613	0.3125
Gaussian	Categorical_crossentropy	6.0312	0.3750
Gaussian	Logcosh	4.2236	0.2708
Gaussian	Hinge	1.1200	**0.6251**
S-shaped	Mean_Absolute	4.3112	0.4565
Sshaped	MSE	4.2674	0.2917
S-shaped	Categorical_crossentropy	5.8769	0.0333
S-shaped	Logcosh	4.4675	0.1667
S-shaped	Hinge	5.5678	0.0000
Triangular	Mean_Absolute	4.2978	0.0833
Triangular	MSE	4.2634	0.2708
Triangular	Categorical_crossentropy	5.7617	0.0208
Triangular	Logcosh	5.5876	0.1875
Triangular	Hinge	5.7665	**0.6250**

TABLE IV
ACCURACY AND LOSS VALUE OF DROP ROW DATA SET AND WITH SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	4.3342	0.0208
Gaussian	MSE	4.2642	0.2708
Gaussian	Categorical_crossentropy	5.8342	0.0627
Gaussian	Logcosh	4.3421	**0.3542**
Gaussian	Hinge	4.7213	0.0000
S-shaped	Mean_Absolute	4.3124	0.2292
S-shaped	MSE	4.2343	0.1875
S-shaped	Categorical_crossentropy	6.2212	0.0625
S-shaped	Logcosh	4.3427	0.2083
S-shaped	Hinge	4.7453	0.0417
Triangular	Mean_Absolute	4.2634	0.1458
Triangular	MSE	4.5674	0.1875
Triangular	Categorical_crossentropy	6.0354	0.0417
Triangular	Logcosh	4.2354	**0.3750**
Triangular	Hinge	4.7123	0.0000

the best value of 64.57%.

VII. CONCLUSION

This paper described a technique that encrypts and hides sensitive information but also sends the data to different organizations securely. In order to encrypt sensitive data, our approach used three different fuzzy logic membership functions.

We kept the boundary points secure. The boundary points are only known to the sender and receiver. Then, we used an autoencoder to learn the input feature vectors of the modified data set which then allows us to send the output of the autoencoder to share data with other organizations. As for the experiments, we evaluated three types of data sets. The data sets were modified by dropping columns, dropping rows and and replacing the missing values with the mean. We then evaluated the accuracy against different loss functions and measured the accuracy of the autoencoder with and without sparsity constraints. From all the results we evaluated, we found that the 'dropping column data set' and running the

TABLE V
ACCURACY AND LOSS VALUE OF MEAN DATA SET AND WITHOUT SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	0.2648	**0.6250**
Gaussian	MSE	4.3532	0.2500
Gaussian	Categorical_crossentropy	2.5200	0.0000
Gaussian	Logcosh	1.3600	**0.6252**
Gaussian	Hinge	2.1500	0.0000
Sshaped	Mean_Absolute	1.3800	0.2086
Sshaped	MSE	1.3600	0.5331
S-shaped	Categorical_crossentropy	2.2000	0.0000
S-shaped	Logcosh	1.3600	0.5993
S-shaped	Hinge	2.1400	0.0000
Traingular	Mean_Absolute	1.3600	0.5397
Triangular	MSE	1.3600	0.5199
Triangular	Categorical_crossentropy	2.2200	0.1192
Triangular	Logcosh	1.3600	0.3543
Triangular	Hinge	2.0900	0.0000

TABLE VI
ACCURACY AND LOSS VALUE OF MEAN DATA SET WITH SPARSITY CONSTRAINTS

Membership function	Loss function	Loss value	Accuracy
Gaussian	Mean_Absolute	1.3777	0.1556
Gaussian	MSE	1.3600	**0.6457**
Gaussian	Categorical_crossentropy	2.9306	0.0000
Gaussian	Logcosh	1.3600	0.2815
Gaussian	Hinge	2.1100	0.0000
S-shaped	Mean_Absolute	1.1382	0.1854
S-shaped	MSE	1.3600	0.4437
S-shaped	Categorical_crossentropy	2.6700	0.0000
S-shaped	Logcosh	1.3600	0.2119
S-shaped	Hinge	2.1300	0.0000
Traingular	Mean_Absolute	1.3770	0.1589
Triangular	MSE	1.3685	0.6159
Triangular	Categorical_crossentropy	2.5670	0.5960
Triangular	Logcosh	1.3693	0.2848
Triangular	Hinge	2.1100	0.0000

autoencoder with sparsity constraints obtained the best accuracy. From all the results, we can say that the best accuracy we obtain is by using the autoencoder with sparsity constraints. Among all the fuzzy set functions, the Gaussian membership function with MSE as the loss function using the data set with the dropped column obtained both a very good accuracy and also a low loss value. Hence, the Gaussian membership function can be used to hide / encrypt sensitive information. Also, to send the data we used an autoencoder with MSE as the loss function and obtained an accuracy of 87.34%.

In summary, as our technique is based on data privacy we kept the boundary conditions of the Gaussian Membership functions protected. Also, for better privacy results the boundary conditions should be only used once. If the hospital has to send data again, they should use different boundary conditions. Thus, the adversary would not be able to guess the correct boundary conditions. Our method can assure patients that their sensitive information will be kept secret, and furthermore, other organizations or data analysts will receive the data that is very similar to the raw data for analysis.

For future work, we should hide the sensitive data with other membership functions and evaluate the accuracy of the data set. In addition, we can use different flavors of the autoencoder for sending data from the sender to different trusted organizations.

REFERENCES

[1] R. Mendes, and P. V. Vilela, "Privacy-preserving data mining: methods, metrics, and applications," IEEE Access 5 (2017): 10562-10582.

[2] W. Du, and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining," In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 505-510), ACM 2003.

[3] J. Brickell, and V. Shmatikov, "The cost of privacy: destruction of data-mining utility in anonymized data publishing," In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 70-78), ACM 2008.

[4] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," In Encyclopedia of Data Warehousing and Mining, (pp. 1005-1009), IGI Global (2005).

[5] T. Dalenius, and S. P. Reiss, "Data-swapping: a technique for disclosure control," Journal of statistical planning and inference, 6(1), pp.73-85,1982.

[6] S. P. Reiss, "Practical data-swapping: the first steps," ACM Transactions on Database Systems (TODS), pp.68–73, 1984.

[7] R. Moore, and A. Richard, "Controlled data-swapping techniques for masking public use microdata sets," Statistical Research Division Report Series (1996): 96-04.

[8] J. Domingo-Ferrer, and J. M. Mateo-Sanz, "An empirical comparison of SDC methods for continuous microdata in terms of information loss and disclosure risk," Proc. Joint ECE/Eurostat Work Session Stat. Data Confidentiality, Conf. Eur. Satisticians, 2001.

[9] D. Defays, and M. N. Anwar, "Masking microdata using micro-aggregation," Journal of Official Statistics 14.4 (1998): 449.

[10] J. Domingo-Ferrer, and J. M Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," IEEE Transactions on Knowledge and data Engineering 14.1 (2002): 189-201.

[11] J. J. Kim, and W. E. Winkler, "Masking microdata files," Proceedings of the Survey Research Methods Section, American Statistical Association, 1995.

[12] W. Fuller, "Masking procedures for microdata disclosure," Journal of Official Statistics 9, no. 2 (1993): 383-406.

[13] S. Pattanayak, and S. A. Ludwig, "Encryption based on neural cryptography," In International Conference on Health Information Science (pp. 321-330), Springer, Cham. (2017, December).

[14] S. Pattanayak, and S. A. Ludwig. "A secure access authentication scheme for multiserver environments using neural cryptography," Journal Of Information Assurance And Security, 13.1 (2018): 56-65.

[15] G. Manikandan, N. Sairam, and M. Sathya Priya, "A new approach for ensuring medical data privacy using neural networks," Biomedical Research 28.3 (2017).

[16] V. V. Kumari, S. S. Rao, K. V. S. V. N. Raju, and K. V. Ramana, "Fuzzy based approach for privacy preserving publication of data," International Journal of Computer Science and Network Security, 8(1), 115-121, 2008.

[17] T. Jahan, G. Narasimha, and C. V. Guru Rao. "A Comparative Study of Data Perturbation Using Fuzzy Logic to Preserve Privacy," Networks and Communications (NetCom2013), Springer, Cham, 2014. 161-170.

[18] Fuzzy Logic - Membership Function, last retrieved in January 2019 from https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_membership_function.htm.

[19] L. A. Zadeh, "Fuzzy sets," Information and control 8.3 (1965): 338-353.

[20] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, 313(5786), 504-507, (2006).

[21] Cervical cancer (Risk Factors) Data Set, last retrieved in January 2019 from https://archive.ics.uci.edu/ml/index.php.

[22] Applied Deep Learning - Part 3: Autoencoders, last retrieved in January 2019 fromhttps://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798.