Investigation of Privacy Preserving Methods for Classification and Clustering of Distributed Data

a thesis submitted for the award of

Doctor of Philosophy
in
Computer Science

by

Latha Gadepaka
(Reg. No. 09MCPC10)

Supervisor
Dr. T. Sobha Rani

Co-Supervisor
Prof. Bapiraju Surampudi

School of Computer and Information Sciences
University of Hyderabad
Hyderabad - 500046, Telangana, India
December 2021

CERTIFICATE

This is to certify that the thesis titled **"Investigation of privacy preserving methods for classification & clustering of distributed data"** submitted by **Latha Gadepaka**, to the **School of Computer & Information Sciences, University of Hyderabad**, Hyderabad, for the award of the degree of `Doctor of Philosophy, in Computer Science`, is a bona fide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Dr. T. Sobha Rani
(Supervisor)
School of Computer and
Information Sciences
University of Hyderabad
Hyderabad – 500 046, India

Prof. Bapiraju Surampudi
(Co-Supervisor)
School of Computer and
Information Sciences
University of Hyderabad
Hyderabad – 500 046, India

Dean
Prof. Chakravarthy Bhagavathy
School of Computer and Information Sciences
University of Hyderabad
Hyderabad – 500 046, India

DECLARATION

I, **Latha Gadepaka**, with Register No:**09MCPC10** here by declare that this thesis titled **"Investigation of privacy preserving methods for classification & clustering of distributed data"** is my Ph.D research work submitted to the School of Computer & Information Sciences, University of Hyderabad, Hyderabad, for the award of the degree of Doctor of philosophy in Computer Science.

Date :

Name: Latha Gadepaka
Reg. No.: 09MCPC10

Signature of the Student:

Signature of the Supervisors:

Acknowledgments

My sincere gratitude to my guide **Prof. BAPIRAJU SURAMPUDI**, who is the best role model & the noble human being. Sir has been giving a strong support being my supervisor & mentor since 2003 (from my M.Tech). I am blessed being your student and I am great full and thankful to you sir.

I humbly thank our Dean, SCIS, ***Prof. Chakravarthy Bhagavathi***, for allowing me to re-register and supporting in submission of my Ph.D work. I thank my supervisor, madam ***Dr. T. Sobha Rani***, for supporting in process of my thesis submission. I thank madam ***Prof. P. N Girija*** who was my DRC member and I thank my present DRC members ***Prof. Abdul Salman Moiz*** and ***Dr. Nagender Kumar S.***, for their valuable inputs. I specially thank madam, ***Dr. Durga Bhavani. S*** for the moral support from the beginning.

My heartfelt gratitude & thanks giving to my greatest parents **Deena & Dasu Gadepaka**, for their unconditional love & support in every point of my life. Thank you Mom and Dad for your wonderful brought up with your valuable teachings, with many sacrifices, struggles and continuous prayers. I am being blessed as your youngest daughter.

My sincere gratitude to my in-law parents ***Pulla Jayapal Raj*** garu & ***Susheela*** garu, for the immeasurable support in taking care of my little boy, while I am working and writing my thesis. I extend my gratitude to my brother in-law ***Mahipal Raj*** garu, ***Sudheera*** akka and beloved kids ***Abhishek & Abhinav*** for being my strength & support all the time.

Thanking my elder sister ***Rani*** for being my only support even in her difficult times (missing my brother in-law ***Rajasekhar*** garu who encouraged me well). Thanking dearest kids ***Ricky & Chicky*** for being part of my joyful times & funny fights too. Thanking my dear caring sister ***Shyama Latha Devi***, for being my strongest motivational support all the time. Thanking my only beloved brother ***Swarna Kumar Gadepaka*** for encouraging & supporting me, and thanking my sister in-law ***Shailaja*** for giving moral support. Thanks to dearest kids ***Goldy & Candy*** for being part of my happiness.

Thanks to my dear friend ***Rajini*** for being always courageous and special thanks to ***Vinuthna*** for giving a helping hand in right time. Special thanks to dear pastors ***Rev. Charles Theodore*** garu, ***Rev Dr. Purushotham*** garu and all the church members for their prayer support.

––Thanking you all––

Dedicated To..

To My Beloved Husband..
Mr. Manipal Raj Pulla
&
To My only Son..
Michael Melchizedek Levinpal Raj

Abstract

Privacy preserving data mining has been a big research domain with lot of progress in developing secure algorithms for various data mining tasks such as classification and clustering. It is important to address the privacy of each individual when the data is distributed among different parties to get general outcomes. There has been lot of progress in privacy preserving methods in distributed data environment to provide privacy to the sensitive information of a user, even when data is to be partitioned and distributed among multiple parties. It is observed that application of privacy preserving techniques in computational intelligence perspectives like neural network learning, fuzzy logic based learning, and ensemble learning are still open. In view of the increased amount of data every day and privacy concerns of the distributed data, the necessity of preserving privacy has been a must to be addressed and solved while designing algorithms in combined models. This work presents three different methods, Privacy Preserving SOM clustering (PPSOM), Privacy Preserving Fuzzy C-Means Clustering (PPFCM) and Privacy Preserving Global Random Forest Classification (PPGRF), aimed for the privacy preserving in distributed data environment.

The work shows that when data is distributed between multiple parties how the model is capable of preserving privacy of sensitive information of all the parties participating in the computing process. Each problem gives solution for horizontal and vertical data distributions. All the methods and algorithms proposed in this work have shown good performance in the form of privacy, data quality and complexity on example machine learning benchmark data sets such as IRIS, Glass, Wine, and Seeds. In the PPSOM Clustering experiments are conducted using both perturbation based approach and cryptography based approach over horizontally and vertically distributed data among multiple parties. In PPFCM, sequential collaboration when exchanging internal outputs between parties was implemented to perform collaborative clustering and the algorithm has been adapted both for horizontal and vertical data distributions. In PPGRF, a two-phase approach that combines local random forest with global random forest was utilized to build the final global random forest by aggregating based on voting between parties. Again, the proposed PPGRF algorithm is adapted for horizontal and vertical data distributions. The main aim of the thesis of developing solutions for preserving the privacy has been successfully achieved with very less data loss and acceptable computational cost. In addition, the thesis presents assessment of privacy level of distributed data using privacy metrics.

Keywords:*Privacy, PPDM, Perturbation, Distribution, Partitioning, SMC, Cryptography, Clustering, PPSOM, PPFCM, Collaboration, Classifica-*

tion, Ensemble Learning, PPGRF

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Privacy Preserving of Data

Privacy: The privacy is defined when it concern to individually identifiable data, is the protection from an unauthorized intrusion [38]. If the authorization is given to users or data miners to access the data for a purposeful data mining task, then there will not be any privacy issue. In general privacy issue arises at the time of data disclosure and that be viewed in two ways.

- Data is protected form disclosure: Limiting the ability to infer the values from results or even to control the results.
- Indirect disclosure of data: disclosing the data indirectly without violating privacy.

Privacy Preserving: Preserving Privacy of an individual's private data when data is to be distributed among multiple parties in order to get combined results [38]. There are an increased number of methods in Data mining and Information security communities that are addressing privacy and security issues and providing the better solutions. Privacy issues mostly being addressed while extracting or exchanging the data between multiple databases with the goal of getting combined outcomes and protecting privacy of an individual simultaneously [2].

1.2 Privacy Preserving Data Mining

Data mining is the most effective method, that has ability of extracting and analyzing the data from large databases. Once the data is extracted the

data analysis is performed by using data mining functionalities like classification, association rules, prediction and clustering etc. Privacy preserving data mining (PPDM) domain is addressing the most important privacy issues in distributed data environment. As per the literature survey conducted for this work, various articles clearly presented the privacy preserving methods used for preserving privacy of data[1][28]. In some of the published papers of distributed data domain, they addressed privacy preserving mechanisms and models. Agrawal and Srikant [1] proposed solutions by using randomization process which includes the random noise addition to the source data and applying privacy preserving algorithm to further maintain data privacy. Lindell and Pinkas [28] used cryptography based protocols to efficiently and securely build a decision tree and could achieve secure computations.

1.3 Privacy Preserving Distributed Data Mining

In distributed data environment, when the data is distributed among two or more number of parties, then "No party should know anything more than its own input and a prescribed output and the only knowledge a party should learn or know from other party is only the output of other party [27]. The main objective of any data mining technique is to extract generalized knowledge from dataset rather than extracting individually identifiable information. When we observe in most of the distributed data mining applications, the data can be shared and accessed among multiple organizations(locations/sites), even though the data is under authority of the data owner. In distributed data environment there is always a scope of privacy violation by determining the personal information of an individual, that leads to serious data loss and misuse of data. Hence the privacy preserving data mining algorithms and models to be effectively designed to ensure that the knowledge is extracted for a right purpose and the privacy is preserved at the same time. Privacy preserving distributed data mining domain is providing some possible solutions by making use of the methods like, Data swapping technique, data perturbation by random noise addition, oblivious polynomial transfer, data anonymity approach and random projection based approaches etc. These are some ways of secure computational methods being successful in producing combined results without effecting the privacy of an individual while exchanging their information among multiple parties.

1.3.1 Privacy Concerns of Distributed Data

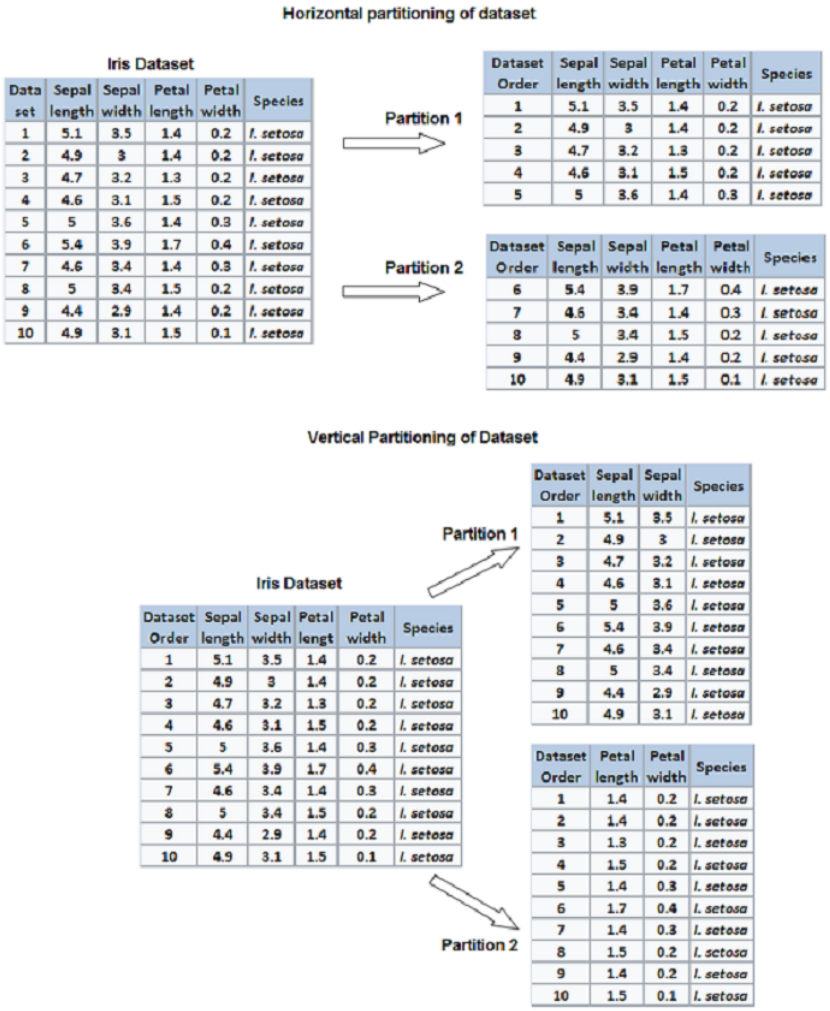
Addressing the privacy issues of distributed data has become the most important task. The privacy of an individual must be protected from violation when the data is distributed among different parties to get general outcomes. There has been lot of progress in privacy preserving methods in distributed data environment to provide privacy to the sensitive information of a user, even when data is to be partitioned and distributed among multiple parties. For example in medical research multiple databases like medical data, patient data, medicines data and purchase data etc to improve business with consumers. Data exchange or sharing of data is necessary for making use of data resource mangers or data miners in order to get combined outcomes for purposeful decision making. But the data may have some private information that cannot be shared, hence privacy concerns or issues must be addressed in order to protect the data and its privacy.

If data mining techniques to be applied in medical research in order to study and analyze the health related problems, then that requires extracting information from various medical centers or hospitals and also from patients. In such situations the privacy of patients must be preserved as per the Health Insurance Portability and Accountability act (HIPPA) [18]. The another act named "Data-Mining Moratorium Act" which is introduced by defense department of USA, banned all data mining operations including research and development due to expected privacy violations [13]. Hence the privacy preserving must address al the privacy issues and assure the privacy of data at the time of computing combined results for decision making. Some of the existing privacy preserving methods and models are presented in a summary table in appendix.

1.3.2 Data Partitioning Methods

In privacy preserving distributed data mining environment, how the data is being partitioned and shared between parties will effect the way of privacy preserving. In distributed computing environments data set can be partitioned into multiple partitions as per the requirement. A dataset is divided and distributed into multiple partitions in order to get common outcomes in shortest communication time. In distributed data domain there are different ways of partitioning methods existing. We adopt and use horizontal way of partitioning and vertical way of partitioning in this thesis work as shown in below diagram for example data set (iris).

Figure 1.1: Horizontal and Vertical Partitioning of Dataset



A data set is defined as $D = (E, I)$, where E is a set of entities for which the required information to be collected and I is a set of features collected for an entity set.

Horizontal Partitioning : in horizontal way of partitioning the same type of information is gathered for different set of entities. Here different parties from differect locations can extract same set of features for different entities. Each party owns collection of horizontal instances (rows).

Vertical Partitioning : in vertical way of partitioning different type of information will be extracted for same set of entities. Each party owns set of vertical instances (columns).

1.3.3 Perturbation based Privacy Preserving

The main objective of data perturbing is not to reveal original input information to any data mining algorithm or model. In distributed data environment if combined results to be obtained from multiple data sets without violating privacy of an individual is possible with perturbation based method. Since the data doesn't send the exact (original) input values, then if that input consists any sensitive data item inked to an individual, will be protected from unauthorized disclosure. There are two ways of data perturbation, additive perturbation and multiplicative perturbation. The additive perturbation method is used to perturb the sensitive or private data values by adding some random noise to the original value and share the perturbed data instead of original data and also assure the privacy while valid output is securely obtained [38]. If $X{=}x$ is an individual attribute that represents information of an individual, then x will be added with a random number r generated form a normal distribution. Now the perturbed value of $x{=}x + r$, will be shared instead of original value x to the data miner. Following table shows an example representation of original values after data perturbation is applied.

Figure 1.2: Perturbed values of original Data

No.	Age	YearsEdu	Income (Original)	Income (Perturbed)
1	25	16	54	57.0
2	31	14	55	52.0
3	32	18	60	57.0
4	36	12	49	52.0
5	43	16	65	61.3
6	48	20	70	71.5
7	50	13	57	61.3
8	53	18	73	71.5
9	56	14	62	61.3

The challenging objective of perturbation technique is to get considerable results for different data mining tasks. Agrawal and Srikant could derive the first solution for this type of problems where, the small random noise is added to the original data value. Then a new dataset with randomized or perturbed data set is created for using in privacy preserving methods. They came up with acceptable similarity while reconstructing distribution of perturbed data and the distribution of original data.

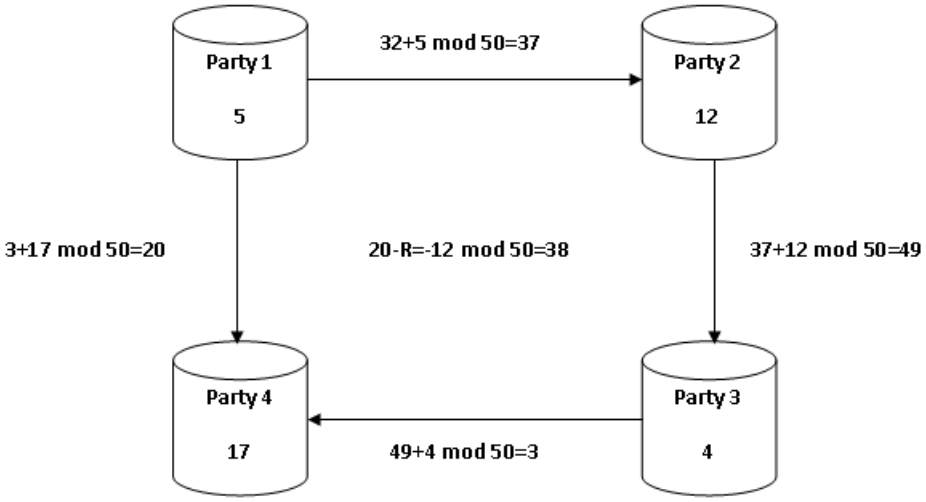
1.3.4 Cryptography-based Privacy Preserving

In cryptography based solutions for privacy preserving, algorithms are designed as protocols that are able to share the input to any other party in an encrypted format. When ever the information is to be accessed by the other party that must follow the decryption of input information with mutual understanding between parties. In the beginning Secure Multi Party computing was introduced for decision tree classification using ID3 algorithm by Lindell and Pinkas[27]. A cryptography based protocol was designed and proposed by Du and Zhan to preserve privacy using ID3 algorithm over vertically partitioned data [42]. Chris Clifton proposed secure computing method for clustering using Expectation Maximization algorithm over vertically partitioned data[27]. Kantarcioglu and Clifton proposed some protocols for privacy preserving distributed data mining of association rules on horizontally partitioned data [21]. It is observed that there is always a communication overhead on the model when cryptography based protocols are used for data exchange between parties.

1.3.5 Secure Multi Party Computing

In distributed data environment if two or more parties are involved to get combined results, then privacy issues to be considered and see that whether the privacy of any party is being violated while producing the combined results. Secure Multi party computing describes secure computation of a result (sum or product etc) with distributed inputs where two or more parties are involved in giving inputs and producing combined outputs [38] [**?**]. The ultimate goal of this SMC approach is without disturbing privacy of an individual, the expected results are to be produced. YAO's protocol [40] is used in SMC where Sum or Product of numbers are calculated at various parties using a secure MOD function, into which the inputs are given by all the parties until the final output has been adopted by the first party where the process has been started. Privacy preserving data mining has solutions using Secure multi party computing and most of the cryptography based solutions are referred to be SMC based solutions. The example is given to understand the process of securely calculating the sum of two numbers using SMC.

Figure 1.3: Computing Secure Sum using Secure Multiparty Computing



In this method initially secure two-party computation was proposed Yao, where he gave the solution for millionaire problem by using secure two-party computation [40]. In this problem two millionaires could find out who is richer than the other without revealing their exact amount of wealth. Then secure multi party computing is introduced where most of the privacy-preserving problems are addressed and solved. In SMC based approaches there are some models for data sharing and secure computing in order to protect privacy of multiple parties. Three of them are explained as follows.

- **Trusted Third Party Model:** In information security domain the standard for protecting privacy of data is based on assuming that, there is a trusted third party to whom one can share information. For example if a model has three parties, the third party performs the required common computations and delivers only the outputs or combined results to other two parties. It is assured that except the third party, nobody learns anything from other party except its own input and the combined results.

- **Semi Honest Model:** Semi-honest model is also known as the honest-but-curious model, which follows the rules of the concerned protocol, but after that the protocol is free to use whatever it sees during execution of the protocol to compromise security.
- **Malicious Model:** This model has no restrictions imposed on any of the participant parties, that is any party will be fully free to involve and do whatever the action it want to do in the total process. It is quite challenging to develop efficient protocols under the malicious model, which does not assure privacy for many applications.

1.4 Privacy Preserving Methods & Models : A Literature Survey

There has been lot of research work going on privacy preserving methods applied in computational intelligence domain. This section presents a literature survey on previous work based privacy preserving methods. Work published on different privacy preserving methods like, decision tree classification [22], privacy preserving self organizing map [31], privacy preserving fuzzy based clustering [43] [25] and privacy preserving random forests classification [29] by various authors. In distributed data environment how the privacy is preserved [15]and what are the privacy measures to be used for deriving privacy level explained in different publications [30] [39] [10] [26]. Comparative studies of existing methods also addressed in some of the published works [7]. The work from different research articles motivated to choose and work further for proposing privacy preserving problems and providing solutions. This section has focused on some of the literature helped in exploring our proposed methods in providing solutions in this thesis.

1.5 Thesis Organization

- **Chapter 1 (Introduction)** presents the details of Privacy and the data mining methods of preserving privacy when data is distributed. It has also given the details of data partitioning methods and Privacy metrics.

- **Chapter 2 (Privacy Evaluation)** presents the privacy evaluation criteria with the help of privacy metrics used to measure privacy. A systematic survey is presented followed by the summary of all privacy metrics used for proposed privacy preserving methods of this thesis work.

- **Chapter 3 (PPSOM)** will decribe the privacy preserving clustering using self organizing map for both horizontal and vertical data distributions. It includes the results and chapter summary.

- **Chapter 4 (PPFCM)** will describe the privacy preserving Fuzzy C-Means clustering in collaborative environment. Chapter gives the details of FCM clustering and privacy preserving collaborative FCM clustering follwed with results and chapter summary.

- **Chapter 5 (PPGRF)** will describe the privacy preserving Random forest classification, building the global random forest from number of local random forests. It ends up with chapter summary after presenting the results.

- **Chapter 6 (Conclusions &Future Scope)** gives the detailed thesis summary, limitations of each problem, future scope and directions for future work.

Chapter 2

Privacy Evaluation of Privacy Preserving Methods

Privacy has no unique standard definition, hence quantifying privacy level has been a challenging task. There are some privacy evaluation metrics proposed in the context of Privacy Preserving Data Mining[30]. A privacy preserving method can be evaluated based on the following criteria.

- **Privacy Level:** The level of privacy is assured by a privacy preserving method that indicates how near the privacy of sensitive data is preserved and still original data can be estimated.
- **Hiding Failure:** This examines that whether the private information which is protected by the privacy preserving model successfully or not.
- **Data quality:** The quality of data (originality) is verified before and after applied over privacy preserving technique.
- **Complexity measure:** This measures the efficiency of a privacy preserving algorithm, with respect to the resources that use in the process.

2.1 Privacy Evaluation Metrics

The output of a privacy metric represents a particular property measured by a privacy metric. It is observed that, no single metric is enough to measure the privacy level because of multiple parameters involved in preserving privacy of a method. There are different properties of outputs to represent different aspects of privacy. A complete estimation of privacy for any privacy preserving mechanism can be obtained from different output categories.

This section gives some categories of privacy metrics based on the output properties as per the literature survey[39].

- **Uncertainty Metrics:** This metric measures the privacy level assuming high uncertainty in other party's estimation, on information known with certainty related with high privacy (other party can not depend upon the guess made from information known with certainty). An other case could be individual users may lose privacy even when the other party guesses correct having highly uncertain information.

- **Information Gain/Loss Metrics:** These metrics are based on gain or loss of information at the time of exchange of information between multiple parties. The metric measures the amount privacy lost or information received by other party at the time of data disclosure.

- **Data Similarity Metrics:** The data similarity is measured within a data set or between multiple data sets. The similarity could be frequency of attributes or numerical values of attributes.

- **Indistinguishability Metrics:** measures do not quantify the level of privacy but, provides a binary indication on whether two outcomes(of Party-1 and Party-2) of a privacy mechanism are indistinguishable (different/distinct from each other) or not. Privacy is high if it cannot be distinguished between any pair of outcomes.

- **Probability of success metrics:** This type of metric quantify the privacy based on the probability of succeed in single attempt and multiple attempts. Low success probabilities correlate with high privacy. Individual user may still suffer a loss of privacy even when the other partys success probability is low.

- **Error based metrics:** measures how correct the other partys estimate is (eg. distance between the true outcome and estimated outcome). High correctness and low errors correlate with low privacy.

- **Time-based metrics:** measures (a) time of other partys success (a longer time relates with high privacy) or (b) time until confusion (shorter time relates with higher privacy).

- **Accuracy/Precision Metrics:** quantify how precise the other party's estimate is without considering the correctness. The more precise estimates relates with lower privacy level.

2.2 Privacy Evaluation Process

Once the privacy preserving mechanism, procedure, technique or method is implemented and the results are published, then the privacy concerns addressed in algorithm to be proved with respect to the privacy level it has reached. The input for privacy metrics is the results of privacy preserving method. The privacy evaluation metrics used for proposed methods in thesis are explained in this section to give the route map for presenting results of privacy metrics of proposed methods (PPSOM, PPFCM and PPGRF).

2.2.1 Privacy Evaluation of Privacy Preserving Self Organizing Map

Normalized Variance: In perturbation based privacy preserving methods a normalized variance is derived from the statistical variance σ^2 and the dispersion between original data X^* and perturbed data Y is measured. The high normalized variance give the better privacy level.

$$priv_{VAR} \equiv \sigma^2(X^* - Y)\sigma^2(X^*) \tag{2.1}$$

Conditional Privacy Loss: This metric measures the proportion or fraction of privacy of data X^* that has lost by revealing the data Y (data revealed and observed by other party). Low value refers high privacy level.

$$priv_{CPL} \equiv 1 - 2^{I(X^*;Y)} \tag{2.2}$$

Positive Information Disclosure: Quantifies the prior probability of the private input (perturbed input) X^* and posterior probability of a new observation y (output) and check for the equality. Low probability of disclosure indicates high privacy level.

$$p(x^*) = p(x^*|y) \tag{2.3}$$

2.2.2 Privacy Evaluation of Privacy Preserving Fuzzy C-Means Clustering

Cluster Similarity: A clustering algorithm is applied to series of transitions for original data T_{X^*} and transitions for modified data T_Y. The cluster similarity is measured between two sets of clusters C_{X^*} and T_Y belongs to original and modified data respectively. Miss placed transitions are identified by computing element wise subtraction between two clusters. The percentage

of correctly clustered transitions refers the privacy level of original hidden values.

$$priv_{CS} \equiv 1 - \frac{|\forall_i : C_{Yi} - C_{X^*i}|}{|T_{X^*}|} \tag{2.4}$$

t-Closeness: The earth mover distance d must be smaller than a threshold value $'t'$, between distribution S_E of sensitive attribute values in a class E to be closer to their local distribution S.

$$priv_{TC} \equiv t, \forall_E : d(S, S_E) \leq t \tag{2.5}$$

Privacy Score: The risk of a user u increases with the sensitivity ωx^* of information items $x^* \in X^*$ along with their visibility $Vis(x^*, u)$. The more visibility refers the low privacy score.

$$priv_{PS} \equiv \sum_{x^* \in X^*} \omega x^*.Vis(x^*, u) \tag{2.6}$$

2.2.3 Privacy Evaluation of Privacy Preserving Global Random Forest Classification

Cumulative Entropy: A combined zone R is common location where many nodes are close to each other at the same time. Cumulative entropy is summation of all individual entropies $H(X_r)$ of combined zone r. The high entropy value indicates high privacy level.

$$priv_{CUE} \equiv \sum_{r \in R} H(X_r) \tag{2.7}$$

Conditional Mutual Information: Quantifies the amount of sensitive information and correlation between sensitive data X^* learned by observing Y, for given prior knowledge Z.

$$priv_{CMI} \equiv I(X^*; Y|Z) = H(X^*|Z) - H(X^*|Y, Z) \tag{2.8}$$

Percentage Incorrectly Classified: The percentage of miss classified events U' within the set of all events or users U is measured.

$$priv_{PIC} \equiv \frac{U'}{U} \tag{2.9}$$

2.2.4 Privacy Metrics based on Error, Time & Accuracy of PP Methods

Mean Squared Error: measures error an other party makes in creating his estimate. It is an error between observations y made from other party and the true outcome x^*.

$$priv_{MSE} \equiv \frac{1}{|X^*|} \sum_{x^* \in X^*} ||x^* - y||^2 \tag{2.10}$$

Time until Success: Assuming that the other party succeeds the time until other part'y success is measured. The result depends on the success and varies as per the process flow of privacy preserving method. Example (i) success could be , identifying n out of N targets in multiple parties, (ii) when one party first compromises a communication path.

$$priv_{ST} \equiv Time(n \in N) \tag{2.11}$$

2.3 Privacy Level in Proposed Methods

The table 2.1 shows all the prerequisites to measure privacy of a privacy preserving method and model, that consists of 14 different privacy metrics can be useful for measuring privacy level of algorithms proposed and implemented in this work. The table presents value ranges ([0,1] and [0,∞]) to be considered to decide the privacy level (High/Low), and it also shows the input data source used, stating that whether the input is taken from published, observed and reported from the privacy preserving methods. The table 2.2 presents the results of privacy metrics and privacy levels observed for the privacy preserving methods proposed in this work. The privacy analysis is presented for each problem in individual chapters based on the privacy level observed by privacy metrics presented here.

Table 2.1: Privacy Metrics and their value ranges to measure Privacy Level

Sno	Output Metric	Range	High/Low	Data Source
1	Normalized Variance	[0, 1]	High	published
2	Conditional Privacy Loss	[0, 1]	Low	obs, pub
3	Positive Information Disclosure	[0, 1]	Low	published
4	Cluster Similarity	[0, 1]	Low	obs,rep
5	t-closeness	[0, ∞]	Low	published
6	Privacy Score	[0, ∞]	Low	published
7	Cumulative Entropy	[0, ∞]	High	observed
8	Conditional Mutual Information	[0, ∞]	Low	obs, pub
9	Mean Squared Error	[0, ∞]	High	published
10	Percentage Incorrectly Classified	[0, 1]	High	obs,rep
11	Success Time	[0, ∞]	High	published
12	Mean Time Confusion	[0, ∞]	Low	published
13	Confidence Interval Width	[0, ∞]	High	pub,obs
14	Uncertainty Region Size	[0, ∞]	High	observed

Table 2.2: Privacy Levels of PPSOM, PPFCM & PPGRF measured using Privacy Metrics

PP Method	Privacy Metric	Range	Result	Privacy Level
H-PPSOM	**priv_NVAR**	[0, 1] ($\geq$0.5=High)	0.56	**High**
	priv_CPL	[0, 1] ($\leq$0.5=Low)	0.80	**Low**
	priv_PID	[0, 1] ($\leq$0.5=Low)	0.94	**Low**
V-PPSOM	**priv_NVAR**	[0, 1] ($\geq$0.5=High)	0.93	**High**
	priv_CPL	[0, 1] ($\leq$0.5=Low)	0.81	**Low**
	priv_PID	[0, 1] ($\leq$0.5=Low)	0.24	**High**
H-PPFCM	**priv_CS**	[0, 1] ($\leq$0.5=Low)	0.25	**High**
	priv_TC	[0, ∞] ($\leq \infty$=Low)	1.11	**High**
	priv_PS	[0, ∞] ($\leq \infty$=Low)	1.00	**High**
V-PPFCM	**priv_CS**	[0, 1] ($\leq$0.5=Low)	0.18	**High**
	priv_TC	[0, ∞] ($\leq \infty$=Low)	2.45	**High**
	priv_PS	[0, ∞] ($\leq \infty$=Low)	0.01	**High**
H-PPGRF	**priv_CUE**	[0, ∞] ($\geq$1.00=High)	0.07	**Low**
	priv_CMI	[0, ∞] ($\leq \infty$=Low)	1.00	**High**
	priv_PIC	[0, 1] ($\geq$0.5=High)	0.05	**Low**
V-PPGRF	**priv_CUE**	[0, ∞] ($\geq$1.00=High)	0.95	**Low**
	priv_CMI	[0, ∞] ($\leq$1.00=Low)	0.74	**High**
	priv_PIC	[0, 1] ($\geq$0.5=High)	0.33	**Low**

2.4 Chapter Summary

In this chapter the analysis and summary report has been presented for various privacy metrics for each method used in thesis work. The main purpose of writing this chapter is to introduce several privacy evaluation metrics in measuring privacy level of a privacy preserving algorithm or technique. An analysis of all the privacy metrics along with their respective equations to be used for measuring the privacy level is presented followed by a complete summary table with results of privacy metrics for proposed methods.

Chapter 3

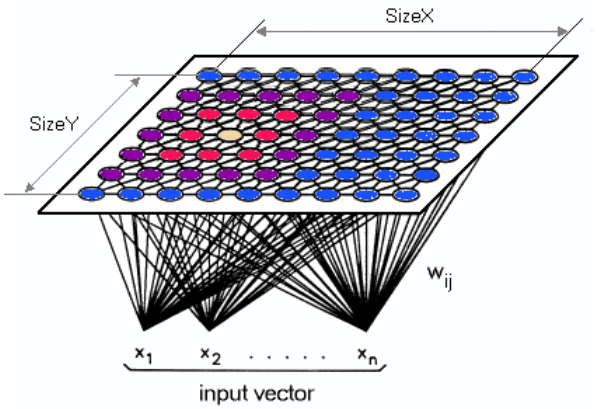
Privacy Preserving Clustering using Self Organizing Map

Preserving privacy in distributed data environment aiming for combined clustering using SOM. Research on privacy preserving methods applied in neural networks are very few like Multilayer Perceptron Learning (MLP), Back-Propagation (BPN) and Self Organising Map (SOM) etc. "Privacy preserving SOM based recommendations on horizontally distributed data" [20], and "SOM-based recommendations with privacy on multi-party vertically distributed data" [19] were well presented by the authors Kaleli and Polat. Their work shows collaborative filtering scheme in SOM clustering estimates truthful predictions while maintaining data owner's privacy on horizontal and vertical data. Their work doesn't follow any specific security method at the input level to preserve privacy of input data where multiple parties are involved. we adopt perturbation based method for horizontal data distribution between parties to do clustering using SOM.

3.1 Self Organizing Map

Self organizing map is a self supervised learning model in neural networks learning, where data is brought into smaller level of groups with similarity in their features through a clustering approach [23]. SOM network also called as topology-preserving map that assumes a topological structure among the cluster units and preserves neighborhood relations and performs topology preserving. The self organizing map provides the better way of mapping between sets of input data items. SOM holds a feed forward structure of nodes, where a single computational layer is determined in rows and columns and each neuron is fully connected to all input layer nodes as shown in figure.

Figure 3.1: Self Organizing Map



3.2 Clustering using Self Organizing Map

In SOM clustering process starts with initializing input and weight vectors then proceeds to determine a winner neuron in competition phase. Then weights are updated for winner neuron and for its neighborhood neurons and continued until there is no change in the feature map. The clustering process is given in SOM clustering algorithm.

Algorithm 1 Self Organizing Map Clustering Algorithm

---

1. Competitive Phase: Winner neuron i derived for given $i$nput vector $\vec{x} = [x_1, x_2, ...x_m]$ with weight vector w_j of neuron j $\{j$=1,2,3,...$l\}$ connected to input vector where l is number of output neurons.

$$i(x) = argmin_j \|x_m - w_j\| \tag{3.1}$$

2. Cooperation phase: The topological neighborhood is determined for winner neuron i.

$$h_{j,i(x)} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right) \tag{3.2}$$

3. Weight updating phase: Winning neuron and its neighbor neurons updates their weights

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)\,(x_d - w_j) \tag{3.3}$$

Repeat until no change in feature map & Terminate

For a specified neural network of $a - b$ architecture, an input vector is denoted as $\{x_1, x_2, ...x_a\}$, weight vector is denoted as $w_i j$ and output vector is denoted as $\{o_1, o_2, ...o_b\}$. The winner neuron determines the topological neighborhood $h_{j,i}$. distance is denoted by $d_{j,i}$, that is the lateral distance between winner and excited neurons j. $w_{j(n)}$ acts as forgetting term to stop when process leads to infinity. η is learning rate that should be decreased gradually with time n.

An interesting aspect of unsupervised systems is competitive learning, in which the output neurons compete among themselves to be activated, with the result that only one neuron is activated at one time, that is called the winner neuron [17]. The learning is allowed only for winning neuron, and the resulting algorithm is referred as winner-takes-all learning method, where competition can be implemented by having lateral inhibition connections between the neurons and those neurons are forced to organize themselves, hence such a network is referred as a Self Organizing Map.

3.3 Perturbation based Privacy Preserving SOM Clustering-Horizontal

In perturbation based methods privacy preserving properties are the results of perturbation, where attribute values of individual entities are perturbed or distorted so that the individually identifiable (private) values are not directly revealed. Hence the privacy of an individual attribute values is preserved as explained in chapter 1. The perturbation is applied on the input attribute "x" by additive perturbation method, where an individual attribute value is distorted by adding a random noise r. The perturbed value of "x" is represented as "$x + r$", that will be disclosed as input instead of "x". In general perturbation technique, an individual does not know the direct input sent by another individual other than the data shred with perturbed values. Perturbation based privacy preserving in SOM clustering for horizontal data distribution is proposed and Horizontal-PPSOM algorithm is given below.

Algorithm 2 Privacy Preserving Horizontal-PPSOM Algorithm

Partition: Horizontally Partition dataset & distribute to Party A & B.
Perturbation: Perturbing data by adding random noise r to each input x.
Initialization: Initialize random weights and input parameters.
For all training samples: $\{x_A, x_B\}$
Step 1 : Competitive Phase:
(1.a) For Each output layer node o_i Party A computes

$$\sum_{j \leq m_A} \left(x_j - w_{ij}^o\right)^2 \tag{3.4}$$

Party B computes

$$\sum_{m_A \leq j \leq m_A + m_B} \left(x_j - w_{ij}^o\right)^2 \tag{3.5}$$

(1.b) Secure Sum: Computed by Party A and B for each output layer node o_i

$$o_i = o_{i1} + o_{i2} = \sqrt{\sum_j \left(x_j - w_{ij}^o\right)^2} \tag{3.6}$$

(1.c) Winner Neuron: For Each output layer node o_i find winner neuron i with minimum distance among l output neurons.

$$i = argmin(o1, ..., ol) \tag{3.7}$$

Step 2: Co-Operation & Weight Updation:
For Each output layer weight w_{ij}^o, neighborhood function $h_{j,i}$ is computed by a party holding input pattern x_j. If $j \leq m_A$ then Party A holds input x_j and A computes

$$w_{ij}^o \leftarrow w_{ij}^o + \eta h_{j,i}(x_j - w_{ij}^o) \tag{3.8}$$

If $m_A < j \leq m_A + m_B$ then B holds input x_j and B computes

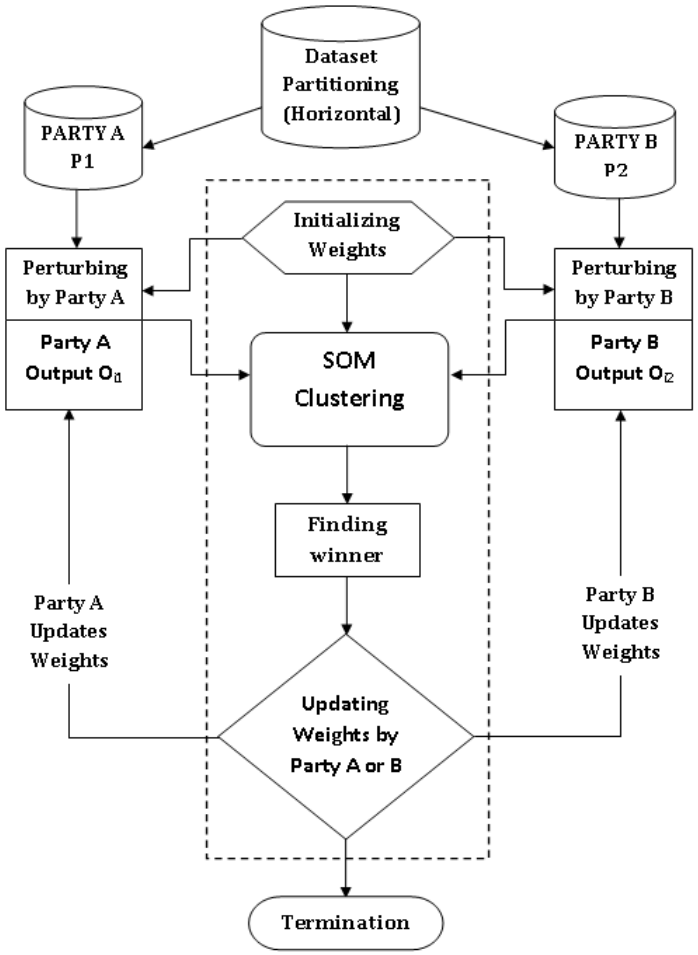
$$w_{ij}^o \leftarrow w_{ij}^o + \eta h_{j,i}(x_j - w_{ij}^o) \tag{3.9}$$

Until Termination Condition

3.3.1 Process of Horizontal-PPSOM Algorithm

In this method the data set is partitioned horizontally and distributed to each party, hence parties hold only few records of entire dataset. Input is shared in a perturbed manner without violating privacy of original data. Process flow diagram is given here followed by the steps in Horizontal-PPSOM clustering.

Figure 3.2: Process flow diagram of Horizontal-PPSOM Clustering



- Partitioning: Partitioning the data set in horizontal style and distribute data set to predefined number of parties.
- Perturbing: Perturb data items, that means adding some random noise generated from a normal distribution to the specified data items of which privacy to be preserved in data set and form a new data set with perturbed values in place of original values.
- Initializing: Initialize all weights and required inputs before start clustering
- Training: Train all the samples located at each party , find winner neuron depending on minimum euclidean distance between nodes at each party

- Weights Updating: Modify weights of winner neuron and its neighbor neurons, then re initialize weights to the network and repeat until all rows of each partition at each party mapped into clusters.

- Termination: Terminate when all data entries of all parties are mapped to any of the clusters, hence Horizontal PPSOM process ends

3.4 Cryptography based Privacy Preserving SOM Clustering-Vertical

When data set is vertically partitioned each party holds only few entity items of an entire record, hence there must be proper security to be applied to protect the privacy of individual information while parties exchange information to compute necessary calculations. The proposed algorithm adopt cryptography based approach of exchanging information between parties where the requires information is shared among parties in a secret manner. In this section we present a privacy preserving distributed algorithm for SOM that follows vertical partitioning and it is composed of one security module, which is used to compute the square root of sum of two numbers securely. The proposed Vertical PPSOM Algorithm is given below.

Algorithm 3 Privacy Preserving Vertical-PPSOM Algorithm

Partition: Dataset is vertically partitioned and distributed to party A & B.
Initialization: Initialize random weights to small random numbers.
Repeat for All Training samples:$\{x_A, x_B\}$
Step1:*Competitive stage:*
(1.1) For Each output layer node o_i Party A computes it's output using m_A

$$o_{i1} = \sum_{j \leq m_A} \left(x_j - w_{ij}^o\right)^2 \tag{3.10}$$

Party B computes computes it's output using m_B

$$o_{i2} = \sum_{m_A \leq j \leq m_A + m_B} \left(x_j - w_{ij}^o\right)^2 \tag{3.11}$$

(1.2) Two parties A and B jointly compute output for each output layer node o_i.

$$o_i = o_{i1} + o_{i2} = \sqrt{\sum_j \left(x_j - w_{ij}^o\right)^2} \tag{3.12}$$

(1.3) For Each output layer node o_i Find the winning neuron i among all l output neurons.

$$Winner(i) \equiv argmin(o_1, ..., o_l). \tag{3.13}$$

Step2:*Cooperation and weight updating stage:*
For Each output layer weight w_{ij}^o, neighborhood function $h_{j,i}$ centered around winning neuron i is computed based on which party holds the input pattern x_j.
If $j \leq m_A$ then A holds the input attribute x_j and A computes

$$w_{ij}^o \leftarrow w_{ij}^o + \eta h_{j,i}(x_j - w_{ij}^o) \tag{3.14}$$

If $m_A < j \leq m_A + m_B$ then B holds input attribute x_j and B computes

$$w_{ij}^o \leftarrow w_{ij}^o + \eta h_{j,i}(x_j - w_{ij}^o) \tag{3.15}$$

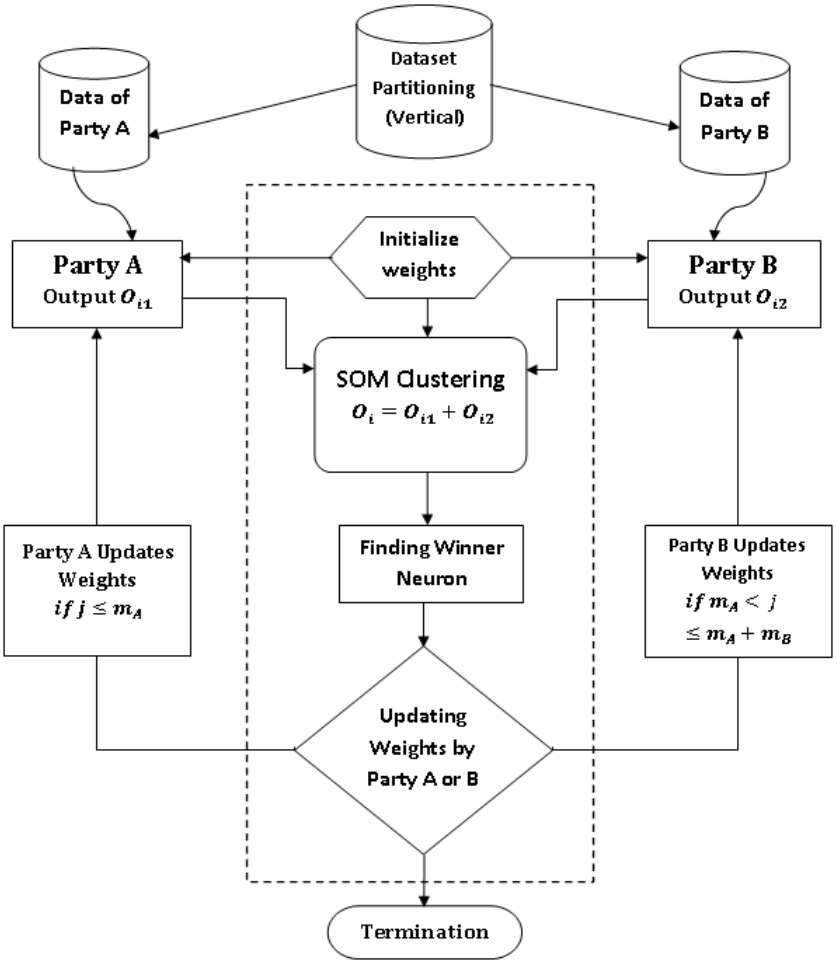
Until(termination condition)

The input of Vertical-PPSOM algorithm is $\langle\{x_A, x_B\}\rangle$ where Party A holds x_A, while x_B is held by party B. The output of Vertical PPSOM algorithm is set of connection weights $\left\{w_{ij}^o \mid \forall j \in \{1, 2, ..., a\}, \forall i \in \{1, 2, ..., b\}\right\}$.

3.4.1 Process of Vertical-PPSOM Algorithm

The main idea of vertical PPSOM algorithm is to secure each step of Non Privacy Preserving SOM algorithm comprises of two main stages, Competitive Stage and Cooperation Stage which include Weight Updating Stage where parties secretly share their individual shares or outputs at each stage. The process flow diagram of Vertical-PPSOM algorithm is presented below followed by steps of the process.

Figure 3.3: Process of Cryptography based privacy preserving in SOM



1. Partitioning: Vertically Partition the data set into predefined number of vertical partitions and distribute between two parties A & B

2. Initializing: Initializes random weights and input parameters

3. Competitive Stage : Individual outputs are computed at Party A & B then winner neuron is declared depending on the euclidean distance equation of algorithm, after outputs are Combined using Secure Sum Algorithm.

4. Cooperative & Weight Updating Stage: Party A & B update their weights privately using neighborhood function, depending on which party holds winner neuron and it's related input.

    - to maitain privacy of their inputs, two parties randomly share their results in encrypted form so that no party able to derive original values of other party.
    - each party holds a "random share", means the intermediate result computed as sum of two random numbers and shares with other party.
    - The process can securely carry forward the random shares in order to compute results and combine results of two parties.

5. Re Initialize updated weights and Repeat the process of Vertical PP-SOM until all data samples located at each party are mapped into any cluster.

6. Terminate the process of Clustering

3.4.2 Securely Computing Sum of Square Root of Two Numbers:

The proof of a secure distributed algorithm (protocol) for computing square root of sum of two numbers is given in this section. In this protocol each party hold some share of the square root and the secure sum algorithm helps them to compute square root of sum of their shares. Considering input x_1 of party A and input x_2 of party 2, the output $\sqrt{x_1 + x_2}$ is securely computed and then shared between two parties [16].

Algorithm 4 Secure sum of square root of two numbers

***S*tep 1:** For Each node i such that $0 < i \leq 2n$, where n is a small integer, a random number R is generated by Party A computes its random share.

$$m_i = \sqrt{x_1 + i} - R. \tag{3.16}$$

***S*tep 2:** Each input m_i is encrypted by party A using *ElGamal* scheme, by adding new random number and gets an encrypted value pair $En(m_i, r_i)$. Party A sends an encrypted value pair $En(m_i, r_i)$ in incremental order of i.
***S*tep 3:** Now Party B choose $En(m_{x_1}, r_{x_2})$ and perform full decryption to get value of m_{x_2}.

$$m_{x_2} = \sqrt{x_1 + x_2} - R. \tag{3.17}$$

R is known only to party A and m_{x_2} is known to only party B. Finally sum of square root of two numbers is computed using

$$m_{x_2} + R = \sqrt{x_1 + x_2} \tag{3.18}$$

3.4.3 Securely Computing Combined Output in Vertical-PPSOM

This section gives steps in securely finding an output at every out put neuron o_i, which is the euclidean distance between input and output neurons in SOM. i is the number of output neurons, j is number of input neurons, x_j is input vector and w_{ij}^o is the weight vector. m_A = party A features and m_B = party B features.

$$\begin{aligned}
o_{i2} + o_{i1} &= \left(\sqrt{\sum_{j \leq m_A} \left(x_j - w_{ij}^o\right)^2 + k} - R \right) + R \\
&= \sqrt{\sum_{j \leq m_A} \left(x_j - w_{ij}^o\right)^2 + k} \\
&= \sqrt{\begin{array}{c} \sum_{j \leq m_A} \left(x_j - w_{ij}^o\right)^2 + \\ \sum_{m_A < j \leq m_A + m_B} \left(x_j - w_{ij}^o\right)^2 \end{array}} \\
&= \sqrt{\sum_{j} \left(x_j - w_{ij}^o\right)^2} \\
&= o_i
\end{aligned}$$

Here o_{i1} is the random share of Party A that is R, and o_{i2} is the random share of Party B. In equation we have k, which is index number of encrypted messages sent by party A and it is computed by Algorithm 3, which is equal to $\sum_{m_A<j\leq m_A+m_B}\left(x_j - w_{ij}^o\right)^2$.

3.5 Experiments & Results:

All the experiments are undertaken for both Horizontal-PPSOM and Vertical-PPSOM algorithms on 4 different data sets taken from UCI machine learning repository [9]. Description of data sets used in experiments is given below.

Dataset	No of Instances	No of attributes	No of Classes
Iris dataset	150	4	3
Glass Identification	214	10	7
Wine dataset	178	13	3
Seeds dataset	210	7	3

Table 3.1: Description of Data sets used in Horizontal PPSOM and Vertical PPSOM

3.5.1 Results of Horizontal-PPSOM

Horizontal PPSOM algorithm is implemented in MATLAB R2013a and compiled with the help of SOM toolbox. The results are shown for 200 epochs, weights are initialized as uniformly random values in the range $[-0.1, 0.1]$ and Learning rate η is taken as 0.1 for all the experiments of Horizontal PPSOM method. The results are presented below for Horizontal-PPSOM Clustering.

Figure 3.4: Horizontal-PPSOM Clustering for Iris Dataset

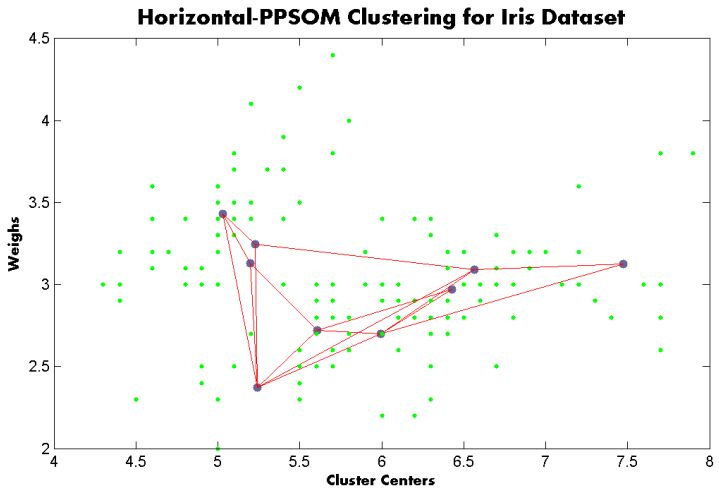


Figure 3.5: Horizontal-PPSOM Clustering for Seeds Dataset

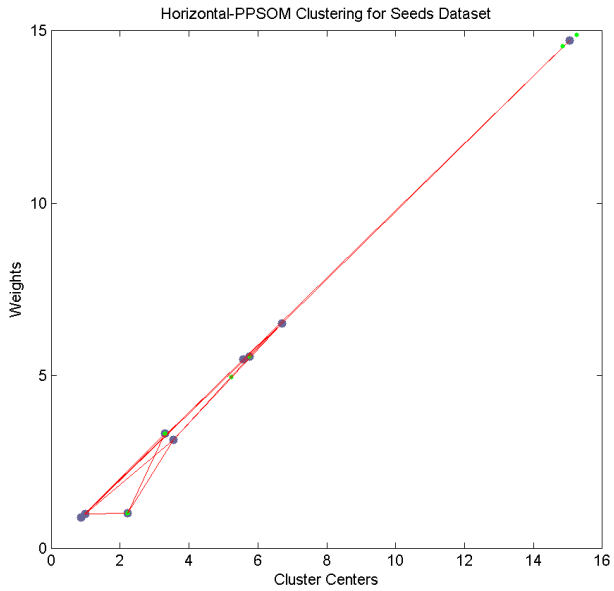


Figure 3.6: Horizontal-PPSOM Clustering for Glass Dataset

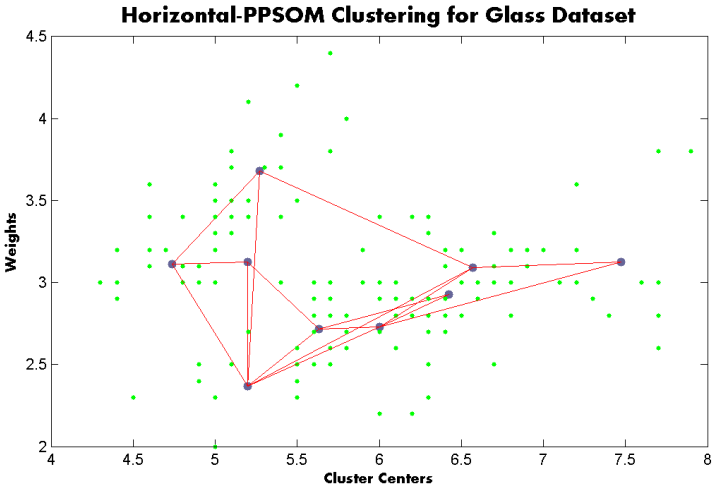
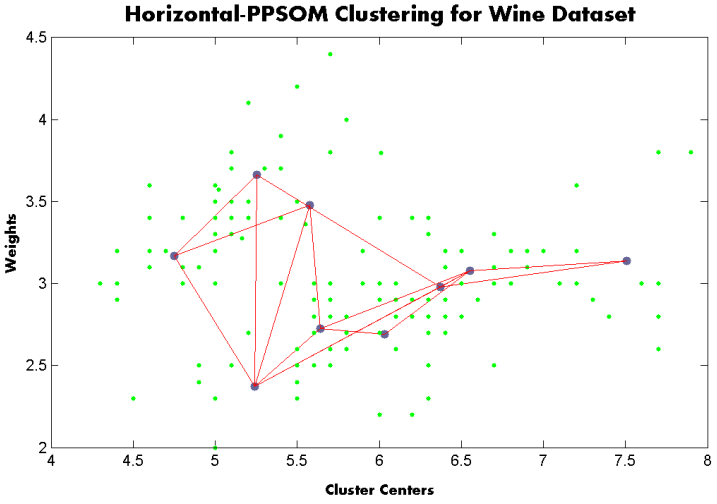


Figure 3.7: Horizontal-PPSOM Clustering for Wine Dataset



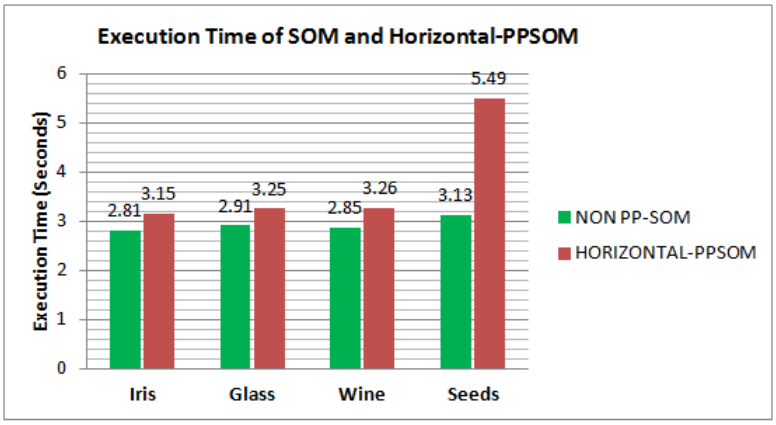
3.5.2 Time complexity and Accuracy of Horizontal-PPSOM

A table and a time comparison graph are given below, for privacy preserving SOM execution time compared with non privacy preserving SOM execution time drawn for four data sets. The results shows acceptable execution times while preserving privacy.

Data set	IRIS	Glass	Wine	Seeds
SOM Execution Time	2.805	2.905	2.853	3.128
Horizontal PPSOM Execution Time	3.146	3.249	3.26	5.487

Table 3.2: SOM execution time (Vs) Horizontal PPSOM execution time

Figure 3.8: SOM Execution Time (Vs) Horizontal PPSOM Execution Time

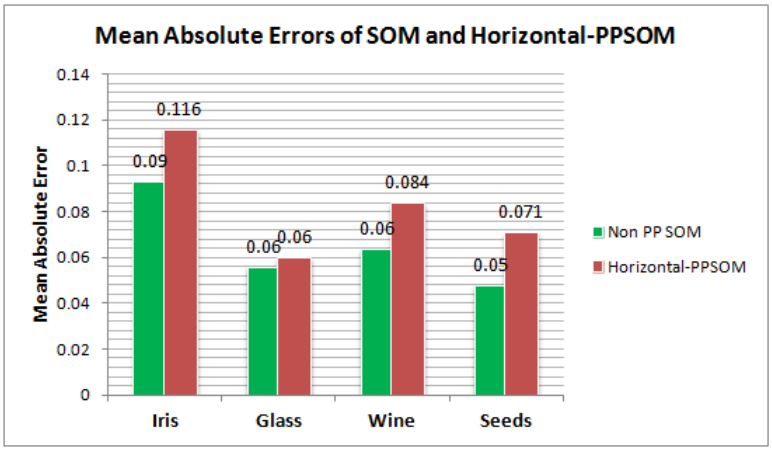


Mean absolute errors observed in SOM and Horizontal-PPSOM for four data sets are given in a table followed by a comparison graph, where an acceptable accuracy has been observed by Horizontal-PPSOM.

Data set	IRIS	Glass	Wine	Seeds
MSE of SOM	0.093	0.056	0.064	0.048
MSE of Horizontal PPSOM	0.116	0.06	0.084	0.071

Table 3.3: Mean Absolute Error of SOM and Horizontal PPSOM

Figure 3.9: Mean Absolute Error of SOM and Horizontal PPSOM



3.5.3 Performance Analysis of Horizontal-PPSOM:

In horizontal partitioning of data, there will be always a flexibility of choosing the number of rows (samples)to be evaluated by the data miner and there will not be any missing of class label, attribute or entity while distributing the data between the parties. The communication cost in privacy preserving methods is always high compared to non privacy preserving methods. If the data is distributed and two or more parties to exchange information then we observe that there is definitely an increased communication cost along with the storage cost of order $O(nm)$.

3.5.4 Privacy Analysis of Horizontal PPSOM:

This is the privacy preserving approach where parties doesn't reveal directly to each other. The only information they communicate or share with each other is the perturbed data and updated weight vector w_j which cannot effect privacy of an individual. The major effect of privacy preserving can be observed when data set is perturbed, partitioned and distributed among various parties involved in the process of clustering. The privacy in Horizontal-PPSOM is analyzed using the metric results given in below table, and privacy issues/concerns when clustering is also explained in view of privacy violation aspects.

- **Input level privacy:***If a party tries to learn target party's input data:* This is the major attack can be happened in any distributed mod-

Table 3.4: Privacy Metrics Results and Analysis of Horizontal-PPSOM

Privacy Metric	Range&Level	Metric Result	Privacy Level
priv_NVAR	[0, 1] ($\geq$0.5=High)	0.26	**High**
priv_CPL	[0, 1] ($\leq$0.5=Low)	0.80	**Low**
priv_PID	[0, 1] ($\leq$0.5=Low)	0.94	**Low**

el. The Horizontal PPSOM algorithm gives the better privacy level measured using privacy metric *priv_NVAR*. When a data set is horizontally partitioned, all the entities and class labels are known to all the parties other than the real values of the samples residing at one party to other party. If one party tries to attack on other's input data then the main privacy preserving method used in our Horizontal PPSOM is Perturbation of the input values before giving them to the next level computations in the model. Hence we prove that when the data is perturbed no party can try to learn or rebuild the original input values given by the owner party. This is applied for all the parties involved in the model aiming for combined result. The privacy level is proved high when the normalized variance between original and perturbed data is high.

- **Process level privacy:***If any party tries to retrieve intermediate outcomes:* This attack can be happened when any party is curious to know the process of computing intermediate outputs of an other party. A party may eager to know the inputs of other party based on intermediate outputs or the weights came from the other party. We prove that there is no way of determining exact values of the inputs even though the intermediate outputs and weight values are known by the other party, because any party gives the output O_{ij} using its own perturbed input values and randomly generated weights. Hence There is no chance of knowing the exact input values even the other party knows the intermediate output values of the other party. In same way the weights come from one party to other are the updated weights after completion of computing the output at that party. Hence the exact weights used by one party to other cant be known.

- **Output level privacy:***If any party tries to predict inputs from the combined output:* This can be done by any party involved in the combined model. We assure the privacy in Horizontal PPSOM at output

level by proving the that any perturbed value at input level can't be reconstructed from the combined output, because of the privacy preserved at input and level and the process level. By any chance if a party tries to learn all the intermediate values and mimics as owner to replicate any of the input and weight values, even then there is no chance of coming up with exact values, because of input level perturbation and regular weight vector modification.

3.5.5 Results of Vertical PPSOM

Vertical-PPSOM algorithm is implemented in Matlab 2013a with the help of SOM tool box. Weights are initialized to uniform random numbers in the range $[-0.1, 0.1]$ and learning rate $= 0.1$.

Figure 3.10: Vertical-PPSOM Clustering for Iris Dataset

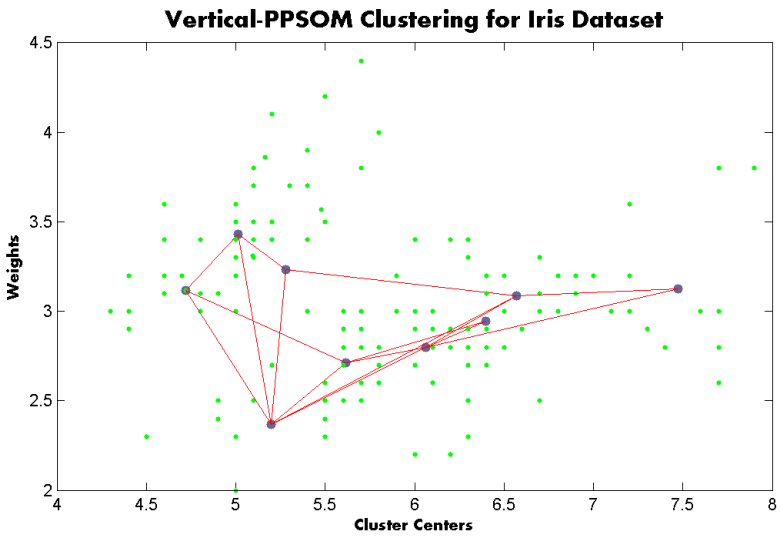


Figure 3.11: Vertical-PPSOM Clustering for Seeds Dataset

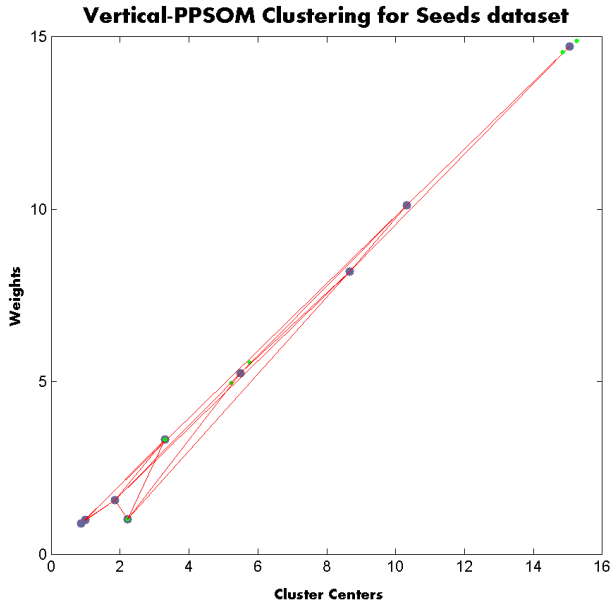


Figure 3.12: Vertical-PPSOM Clustering for Glass Dataset

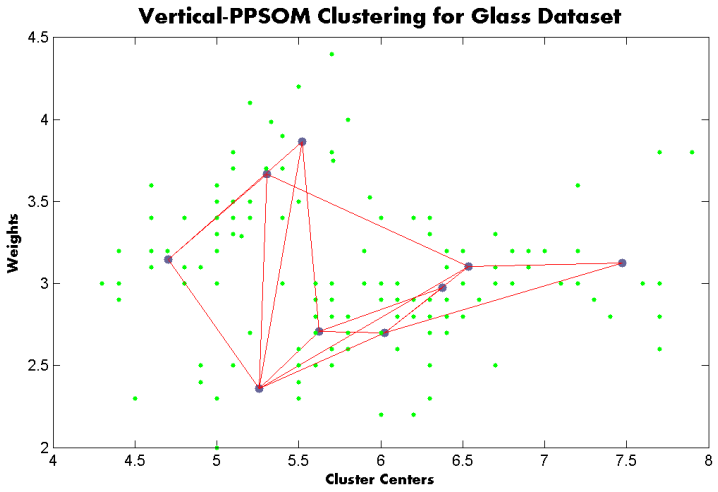
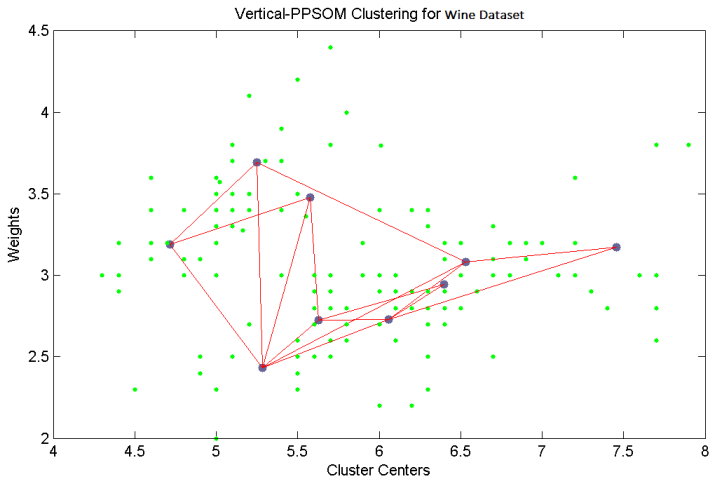


Figure 3.13: Vertical-PPSOM Clustering for Wine Dataset



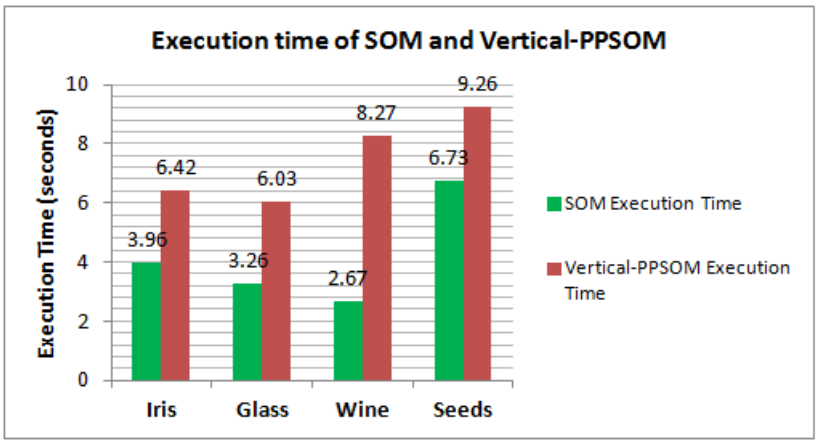
3.5.6 Time complexity and Accuracy of Vertical-PPSOM

A table and a time comparison graph are given below, for privacy preserving SOM execution time compared with non privacy preserving SOM execution time drawn for four data sets. The results shows acceptable execution times while preserving privacy.

Dataset	SOM Run Time	Vertical-PPSOM Run Time
Iris	3.96	6.42
Glass	3.26	6.03
Wine	2.67	8.27
Seeds	6.73	9.26

Table 3.5: SOM Average Time Compared with Vertical PPSOM Average Time

Figure 3.14: SOM Run Time Compared with Vertical PPSOM Run Time

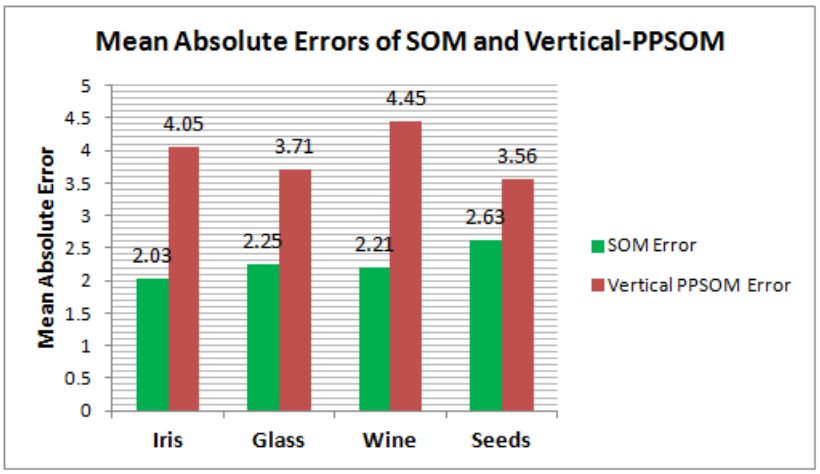


Mean absolute errors observed in SOM and Vertical-PPSOM for four data sets are given in a table followed by a comparison graph, where an acceptable accuracy has been observed by Vertical-PPSOM

Data Set	SOM Error	Vertical-PPSOM Error
IRIS	2.03	4.05
Glass	2.25	3.71
Wine	2.21	4.45
Seeds	2.63	3.56

Table 3.6: SOM Error Compared with Vertical PPSOM Error

Figure 3.15: SOM Error Compared with Vertical PPSOM Error



3.5.7 Performance Analysis of Vertical PPSOM:

Accuracy of our algorithm is satisfactory while preserving privacy. Cryptography based approaches perform encryption and decryption operations for required number of times, which causes high percentage of communication overhead and communication time. Privacy preserving methods may also leads to other added costs like additional storage cost of randomly generated messages by all parties involved in overall process.

3.5.8 Privacy analysis of Vertical PPSOM:

As parties have only few information of the entire record, they exchange intermediate results to each other in order to get a combined output. If there are two parties p_1 and p_2, first p_1 do the necessary calculations to decide the winning neuron and sends its local output to party 2 in a secret way here we use *ElGamal* scheme of public key cryptography with which data can be send to any other party in an encrypted format. Then party 2 receives only encrypted message sent by party 1 then decrypts it based on the keys generated earlier with mutual understanding between parties. Hence we assure privacy in an effective acceptance of performance of the algorithm based on the metrics used to evaluate privacy given in table below. privacy in Vertical PPSOM is explained with respect to the privacy concerns at the expected privacy violation aspects.

Table 3.7: Privacy Metrics Results and Analysis of Vertical-PPSOM

Privacy Metric	Range & Level	Metric Result	Privacy Level
priv_NVAR	[0, 1] ($\geq$0.5=High)	0.93	**High**
priv_CPL	[0, 1] ($\leq$0.5=Low)	0.81	**Low**
priv_PID	[0, 1] ($\leq$0.5=Low)	0.24	**High**

- **Input level privacy:***If a party tries to learn target party's input data:* This is the important aspect of preserving privacy in vertical partitioning algorithms. Only few features/values present at each party and it is always a big challenge when it comes to combine the input and compute combined results in vertical data distribution. The Vertical PPSOM algorithm gives the better privacy assurance compared to the other existing PPSOM algorithms. When a data set is vertically partitioned, all the entities and class labels are un known to all the parties other than the real values of the features residing at each party. If one party tries to attack on other's input data then the cryptography based privacy preserving algorithm assures the privacy of input having used encrypted form of sharing inputs to other party. Vertical PPSOM is assuring the privacy of the input values, before giving them to the next level computations in the model. Hence we prove that when the data is encrypted and distributed, then no party can try to learn or rebuild the original input values given by the owner party. This is applied for all the parties involved in the model aiming for combined results.

- **Process level privacy:***If any party tries to retrieve intermediate outcomes:* This attack can be happened when any party is curious to know the process of computing intermediate outputs of an other party. A party may eager to know the inputs of other party based on intermediate outputs or the weights came from the other party. We prove that there is no way of determining exact values of the inputs even though the intermediate outputs and weight values are known by the other party, because any party gives the output O_{ij} using its own encrypted input values and randomly generated weights. Hence There is no chance of knowing the exact input values even the other party knows the intermediate output values of the other party. In same way the weights come from one party to other are the updated weights after completion of computing the output at that party. Hence the exact weights used by one party to other cant be known.

- **Output level privacy:***If any party tries to predict inputs from the combined output:* This can be done by any party involved in the combined model. We assure the privacy in Vertical PPSOM at output level by proving the that any perturbed value at input level can't be reconstructed from the combined output, because of the privacy preserved at input and level and the process level. By any chance if a party tries to learn all the intermediate values and mimics as owner to replicate any of the input and weight values, even then there is no chance of coming up with exact values, because of input level perturbation and regular weight vector modification.

3.5.9 Complexity Analysis and Scalability of PPSOM:

In both Horizontal and Vertical PPSOM (HPPSOM and VPPSOM), datasets used are with instances up to 300. The computational complexity has shown that, if the number of parties increased and number of secure computations are done in every level of communication, hence the complexity is more when compared to the non privacy preserving SOM clustering. In Horizontal-PPSOM we use perturbation based privacy preserving hence the computational complexity increases for every data instance at the perturbation level. In Vertical PPSOM we use cryptography based approach hence the computational overhead is noticed. We test these methods for a larger dataset (cloud data with 2053 instances and 10 attributes), and could not score acceptable accuracy and privacy level.

3.6 Chapter Summary

We presented Privacy preserving SOM Clustering methods and algorithms for both perturbation based approach and cryptography based approach over horizontally and vertically distributed data among multiple parties. Horizontal PPSOM and Vertical PPSOM are two major algorithms we implemented and presented in this paper. In both horizontal and vertical versions of SOM clustering, our methods are adopted for perturbation based and cryptography based solutions respectively. In general most of the privacy preserving methods compromises with the accuracy, but our model equally ensures the privacy and accuracy. We modified the original SOM algorithm to present Horizontal-PPSOM and Vertical-PPSOM algorithms for both horizontal and vertical versions. The main objectives of proposed algorithms is to securely clustering and computing combined outcomes of two different parties when dataset is horizontally and vertically partitioned. Horizontal-PPSOM and

Vertical-PPSOM could gain considerable privacy level and acceptable communication overhead without violating their privacy. We conclude that both perturbation based method and cryptography based methods we implemented in our algorithms for clustering in privacy preserving SOM provides the better privacy in distributed environment.

Chapter 4

Privacy Preserving Fuzzy C-Means Clustering

In this rapidly growing distributed computing world usually to share sensitive data to others requires strong techniques to ensure privacy of data. The information about people and organizations needs to be shared when a general outcomes are to be collected, hence the privacy concerns of their data must be addressed. The collaborative communication methods and computational schemes are capable of information exchange between multiple parties aimed for combined results[8][14]. Collaborative fuzzy C-Means clustering is an efficient method for discovering combined structure within finite group of different data sites where the data resides at different locations. Fuzzy clustering enhances the efficiency of the model without compromising the privacy at input level of the privacy preserving model. Using fuzzy sets in data mining techniques like clustering, gives good accuracy compared to other clustering methods, as every data point is the member of any of the group with a membership degree[24]. This chapter gives detailed description of collaborative fuzzy c-means clustering in distributed data environment to preserve privacy.

4.0.1 Fuzzy Set Theory

Fuzzy set theory helps to represent the uncertainty, possibility and approximation of a crisp dataset. In general fuzzy logic tries to imitate natural human ability of reasoning. A fuzzy set consists of set of fuzzy membership values u_i ranges in between $[0, 1]$, that are derived by mapping set of real numbers (x_i). A fuzzy membership function can be represented as set of fuzzy values $\{u_1/x_1, u_2/x_2, ...u_n/x_n\}$.

4.1 Fuzzification Method

The ability of converting a given set of numerical data values (crisp data) to fuzzy membership values (degree of membership). For a given universe of X a fuzzy set A is derived with the help of fuzzy membership function $\mu_A(x)$.

$$\mu_A(x) = \begin{cases} 1, & \text{if x in A;} \\ 0, & \text{if x in A;} \\ 0 < \mu_A(x) < 1, & \text{if x is partly in A;} \end{cases}$$

The process of fuzzification and defuzzification of a crisp dataset shown in diagram and an example fuzzy set derived from a crisp set is given below.

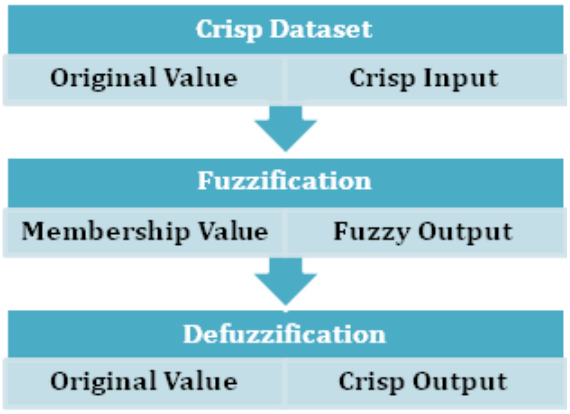


Figure 4.1: The Fuzzification and Defuzzification Process

Table 4.1: Fuzzy membership degree values of a sample crisp dataset

Person Name	Height	Crisp(Boolean)	Fuzzy(Membership Value)
Person 1	205	1	1.00
Person 2	182	1	0.81
Person 3	175	0	0.38
Person 4	167	0	0.10
Person 5	155	0	0.04

4.2 Fuzzy C-Means Clustering

Fuzzy C-Means clustering is a fuzzy based clustering method designed and developed by Dunn in 1973 and enhanced by Bezdek in 1981[6] and has

become efficient tool of analyzing and visualizing data. A fuzzy clustering method devices fuzzy partitions into c clusters, where all the data points are allowed to become member of more than one cluster. The Fuzzy C-Means Clustering brings solution of minimizing degree of membership using an objective function.

$$J_{FCM}(U,V) = \{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m||x_k - v_i||^2\} \tag{4.1}$$

The fuzzy membership function derives the partition matrix $u_{iK} \in [0,1]$ and cluster center v_i for i^th cluster with a non negative fuzzy coefficient value $m = 1$. The fuzzy c-means clustering mainly works with two steps, where in first step optimal membership function will be estimated and in second step estimates the cluster centers. The cluster centers are fixed when ever a membership function is estimated and cluster centers are fixed when membership function is estimated. The FCM follows an iterative procedure as represented in FCM algorithm

Algorithm 5 Fuzzy C-Means Clustering Algorithm

Input: Data set X, No of clusters C and fuzzy coefficient m.
Step 1: Initialize the membership matrix $U = [u_{ij}]$, $U(0)$.
Step 2: Compute Cluster centers $C(k)$ using membership matrix $U(k)$.

$$C_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{4.2}$$

Step 3: Update membership matrix from $U(k)$ to $U(k+1)$.

$$u_{ij} = \frac{1}{\sum_{j=1}^{C}(\frac{||x_i - c_j||}{||x_i - c_k||})^{\frac{2}{m-1}}} \tag{4.3}$$

Continue Step 2 and 3 until termination.

4.2.1 Process of Fuzzy C-Means Clustering

The process of Fuzzy C-Means Clustering algorithm explained in following steps.

1. Computing membership matrix u_{ik} by initializing c centroids and computing membership degrees of elements in order to form clusters, which is computed as a function of closer degrees.

2. Updating the cluster centers as weighted average of degrees of membership to each cluster. The process continues until termination

4.3 Collaborative Clustering

With respect to privacy in distributed computing, the scheme of collaborative computing has been introduced that creates a purposeful communication between different data sites. In collaborative computations of distributed data, the individual data sites to share the information and reconciled in order to get final outcomes. The private local computations can be done at their own sites but to collaborated for final results, which is the best way of maintaining privacy of inputs at each site. The sites to be mutually collaborate and carry forward all the required computations till process is completed [33].

In collaborating approach of clustering there will be two or more number of data sites and at each data site a common structure can be revealed by assuming same number of clusters for each individual data site[7]. In process of collaboration at a particular data site a local level structure of clustering is discovered after receiving the results from other site. The local data site can update the structure according to the data it owns at local level and it's local level findings.

4.3.1 Modes of Collaboration

In collaborative scheme each data site functions like a separate computing entity and restricts to share data outside the site, because of (i) privacy concerns and (ii) feasibility of their technical constraints. Data sites in collaboration exchange only the local outcomes instead of input attribute values. There exists two modes of communication between sites in collaborative approaches as follows [34]:

- (A) centralized mode: in this mode of interaction, considering one data, say D_i , initialized to reconcile the local findings (its local model) with the modeling results available at all remaining datasets $D_1, D_2, , D_i - 1, D_i + 1, .., D_P$.

- (B) distributed mode: in this mode all data sites are allowed to interact between each other and the resulting local outcomes or local models can be shared. Each data site affects each other only when optimizing their parameters based on the local findings of each other.

4.4 Collaborative Fuzzy C-Means Clustering

Collaborative fuzzy C-means clustering was introduced by Pedrycz, motivated by reasonable and useful features of the collaborative computational method applied for fuzzy based clustering. It is observed that the method especially adopted for enhancing quality of unsupervised learning and acquiring better accuracy by using fuzzy based clustering[8]. There are two basic factors that are related to granularity of data in fuzzy based methods, (i) granularity is conveyed by prototypes (cluster centers) (ii) membership degree values are captures by partition matrix (membership matrix). If prototypes are communicated then partition matrix can be developed and if partition matrix is being known prototypes are generated in collaborative clustering. These two important outcomes (prototypes and partition matrix) are the communication sources in collaboration and also privacy preserving channels in complete process of collaboration. The data sets residing at each data site shares the same feature space and a collaboration is established by exchanging required outcomes between two or more data sites. The overall communication is carried through these granular interfaced in collaboration[36]. Structure of a granular interface and their communication in collaboration presented in diagrams below.

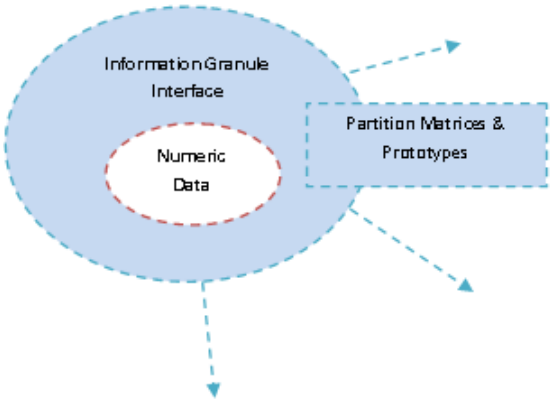


Figure 4.2: The granular interface of the numeric data in collaborative clustering

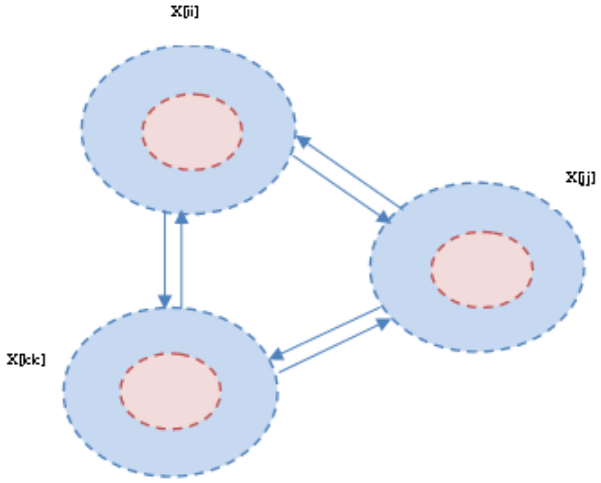


Figure 4.3: The collaboration between granular interfaces of the numeric data

Initially, FCM algorithm is run independently at each data site that happens without any collaboration[41]. After FCM has terminated at each site, processing stops and the data sites communicate their local findings. This communication needs to be realized at some level of information granularity. The effectiveness of the collaboration depends on the way how one data site communicates information granules with another data site [35].

4.4.1 Method of Fuzzy Collaborative Clustering

To explain fuzzy collaborative clustering method carried by different data sites (1,2,...P), first consider an i^th prototype or cluster center at any data site. The sequence of prototypes generated by different data sites (parties) in collaboration is denoted as $v_i[1], v_i[2], ...v_i[P]$. Now, in order to form fuzzy partitions for n dimensional data set over j^th coordinate the collaboration method uses the form $v_ij[1], v_ij[2], ...v_ij[P]$. The splitting is done based on the fuzzy coefficient m, where the values are divided into sets less than v_ij and greater than v_ij. These fuzzy partitioning can be done over horizontal and vertical data distribution and forms corresponding fuzzy partitions $V_{i1}, V_{i2}, ..., V_{in}$ and combined together by taking their Cartesian product.

$$V_i = \prod(V_{i1}, V_{i2}, ..., V_{in}) \tag{4.4}$$

V_i is the granular prototype of the i^th cluster where it's granular information is reflected by reconciling the collaborative outcomes from different data sites.

For a given prototype of i^th cluster derived as result of collaboration, where each membership degree is obtained by the equation given below.

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} (\frac{||x_k - v_i||}{||x_k - v_j||})^{\frac{2}{m-1}}} \tag{4.5}$$

4.5 Privacy Preserving Collaborative Fuzzy C-Means Clustering-Horizontal

Here the privacy preserving collaborative fuzzy C-Means clustering for horizontal data distribution is proposed which is denoted as Horizontal-PPFCM. This process makes use of horizontal collaboration between number of parties involved in collaboration to form specified number of clusters C. When data set is horizontally partitioned and distributed between all parties, then each party holds set of horizontal records and establish a horizontal collaboration. The process preserve the privacy of all parties and collaborative clustering is done without directly exchanging the sensitive information between parties. Communication flow of horizontal collaboration for privacy preserving FCM clustering is presented in below diagram.

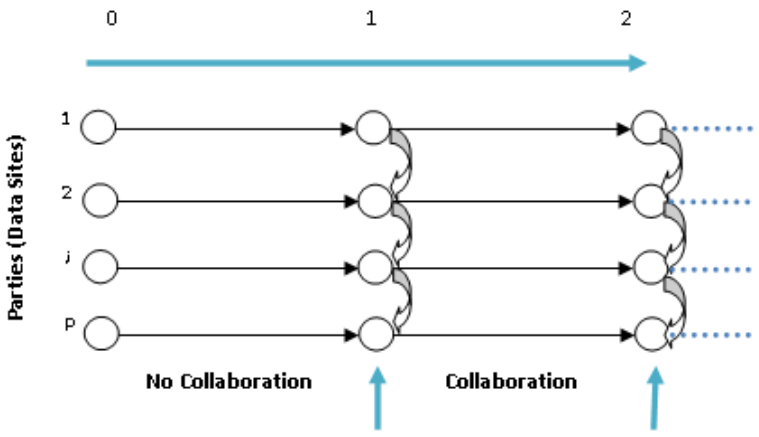


Figure 4.4: Process of privacy preserving collaborative clustering - Horizontal

First the communication path is established and parties are informed about granular prototypes generated at other sites. Each data site does it's own clustering using FCM, while considering structural findings received from other sites, then share the local outcomes to next party. Each site proceeds with its independent computing by considering outcomes communicated by

other parties. All the parties communicate their local findings and set up new constraints for the next phase of the FCM clustering in collaboration process. The next phase of computing is overall collaboration where collaborative computations are carried out and continued until no further change in the complete structure is reported. Algorithm of privacy preserving collaborative fuzzy c-means clustering for horizontal data distribution is given below.

Algorithm 6 Privacy Preserving Collaborative FCM Clustering - Horizontal

Partitioning: Data set is horizontally partitioned and distributed between n parties.

Initializing: Initialize number of clusters c, membership matrix u_{ik}, objective function $Q[ii]$, fuzzy coefficient $m = 2$ & collaboration matrix $\alpha[ii, jj]$.

1. FCM Clustering: For All Parties

The first party p_1 computes prototype v_{ij} vectors for its data and partition matrix u_{ik} for all C clusters, using FCM algorithm.

2. Collaboration Phase:

2.a: Now party p_1 sends it's prototype v_{ij} vectors and partition matrix u_{ik} to next party p_2 through collaboration matrix $\alpha[ii, jj]$

2.b: Now party p_2 computes partition matrix u_{ik} and prototype v_{ij} vectors, then sends to the next party p_n through $\alpha[ii, jj]$.

2.c: Now party p_n computes prototype $V[ij]$ and partition matrix $U[ij]$, then minimizes index of final collaborative function $Q[ii]$ using

$$Q[ii] = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^2[ii]d_{ik}^2[ii] + \sum_{jj=1,jj\neq ii}^{P} \alpha[ii, jj] \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik}[ii] - u_{ik}[jj])^2 \quad (4.6)$$

Repeat Until Termination (End of Collaboration)

4.5.1 Process of Horizontal-PPFCM

Data set is horizontally partitioned and distributed between n parties Initial Party (IP), assigns all its members to the clusters using FCM and the process moves to the next party with updated prototypes of the first party. Next Party 2 do same fuzzy c means clustering as party 1 did, until all its members are allocated to a cluster, then sends new updated prototypes to the next party. After all parties assign their members to the clusters, party n computes and returns the final results through collaboration matrix. The process flow

diagram of privacy preserving collaborative FCM clustering for horizontal data distribution is presented here.

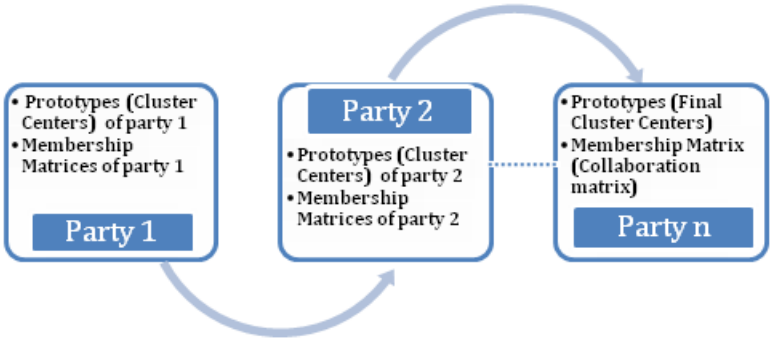


Figure 4.5: Process flow of Privacy Preserving Collaborative FCM Clustering-Horizontal

4.6 Privacy Preserving Collaborative Fuzzy C-Means Clustering-Vertical

The vertical way of collaborative clustering deals with vertical partitioning of data set, where a data set is vertically partitioned and distributed to parties to participate in collaboration. Each party holds disjoint subsets of patterns and these patterns are joined in process of collaboration for finding local outcomes. These disjoint patterns will be in same feature space and commonly adopted for collaborative clustering to join local findings. Vertically distributed mode of collaboration is presented in below diagram.

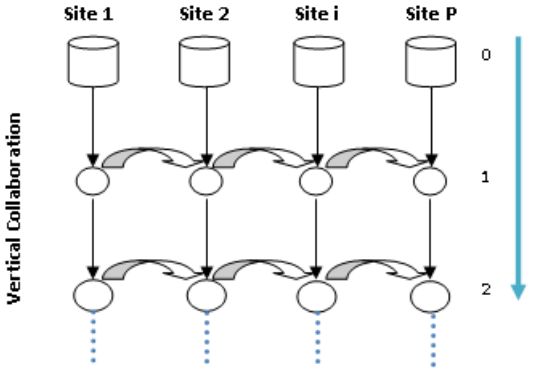


Figure 4.6: Process of Privacy Preserving Collaborative Clustering - Vertical

Here the collaborative communication carried further depending on prototype level where as same objective function and same fuzzy coefficient are used for individual prototypes[12]. The collaborative FCM clustering algorithm proposed for vertical data distribution is given below. In this process

---

Algorithm 7 Privacy Preserving Collaborative Fuzzy C-Means clustering Algorithm - Vertical

---

Data Partitioning: Data set is partitioned into P number of vertical partitions and distributed between $1, 2, ...P$ parties.

Initializing: Number of clusters C, Objective function, fuzzy coefficient $m = 2$ and collaboration matrix $\beta[l, m]$.

FCM Clustering:(Phase-I)

Step-1: For all Parties holding subset of patterns $X_1, X_2, X_3, , X_p$, Randomly initialize partition matrices $U[1], U[2], ...U[P]$.

Step-2: Compute prototypes $V_{i1}, V_{i2}, ...V_{il}$, and Partition Matrices $U_{i1}, U_{i2}, ...U_{ik}$.

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} (\frac{||x_k - v_i||}{||x_k - v_j||})^{\frac{2}{m-1}}} \tag{4.7}$$

Collaborative Clustering: (Phase-II)

Collaboration Matrix: Prototypes $V_i[l]$, partition matrices $U_i[k]$ and collaboration matrix $\beta[l, m]$ combine to minimize objective function $Q[ii]$.

$$Q[ii] = \sum_{k=1}^{X[l]} \sum_{i=1}^{c} u_{ik}^2[l] d_{ik}^2[l] + \sum_{m=1, m \neq l}^{P} \beta[l, m] \sum_{i=1}^{c} \sum_{k=1}^{X[l]} u_{ik}^2[l] ||v_i[l] - v_i[m]||^2 \tag{4.8}$$

Until termination.

---

the prototypes $v_i[1], v_i[2], ...v_i[j]$ of i^th cluster and corresponding membership matrices $u_i[1], u_i[2], ..., u_i[P]$ are computed and joined through a collaborative matrix $\beta[l, m]$. The objective function also referred as collaboration index $Q[ii]$ is used to compute final outcome for l^th data site. All parties in sequence collaborate to generate prototypes and partition matrices, then combine them on collaborative objective function. The collaboration index to be minimized and process terminates once all the elements get into C number of clusters.

4.6.1 Process of Vertical-PPFCM Algorithm

1. For all the parties from $1, 2..P$, of their respective data sites $D[1], D[2], ...D[P]$, define number of clusters C, collaboration coefficient β that optimizes the level of collaboration.

2. **Initial phase:** for each individual data site compute fuzzy c-means clustering results individually in the form of prototypes $v_i[ii], i = 1, 2, ..., c$.

3. **Collaboration phase:** allow the individual results (prototypes)to interact to form collaboration between the sets.

4. For each data site the objective function $\beta[l, m]$ is minimized by considering prototypes and partition matrix communicated by other sites.

5. the process iteratively continued until termination condition of the collaboration.

Privacy preserving model and steps involved in vertically collaborative FCM clustering is given here.

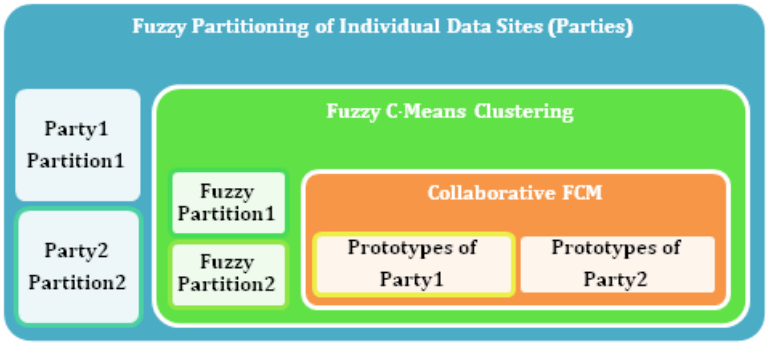


Figure 4.7: Privacy Preserving Collaborative FCM Clustering model - Vertical

4.7 Experiments & Results

All the experiments are undertaken for horizontal and vertical collaboration for fuzzy c means clustering using Matlab2013a with help of fuzzy clustering tool box. The results are presented for collaboration (preserving privacy) and non collaboration (non privacy) based fuzzy clustering for Iris, Glass Identification, Seeds and Wine datasets. For all the experiments, fuzzy coefficient $m = 2$ and the number of clusters are $C = 3$.

4.7.1 Results of Horizontal-PPFCM

The results of collaboration in Horizontal-PPFCM are presented for four datasets. The following figures shows the individual performance of the objective function of FCM and performance of the Collaborative objective function of PPFCM clustering for datasets mentioned earlier (Iris, Glass Identification, Seeds, wine). The performance of objective function is plotted and compared between non collaborative (non privacy) and horizontal collaborative (privacy preserving) fuzzy C-Means Clustering.

Iris Dataset: The first two figure of this section shows the results of non collaborative FCM and the performance of the objective function for iris dataset. The next two figures shows the results for Horizontal-PPFCM when the iris dataset is horizontally partitioned, where each party holds 75 samples each. Figures represents the collaborative clustering of iris dataset along with the performance of the collaboration function.

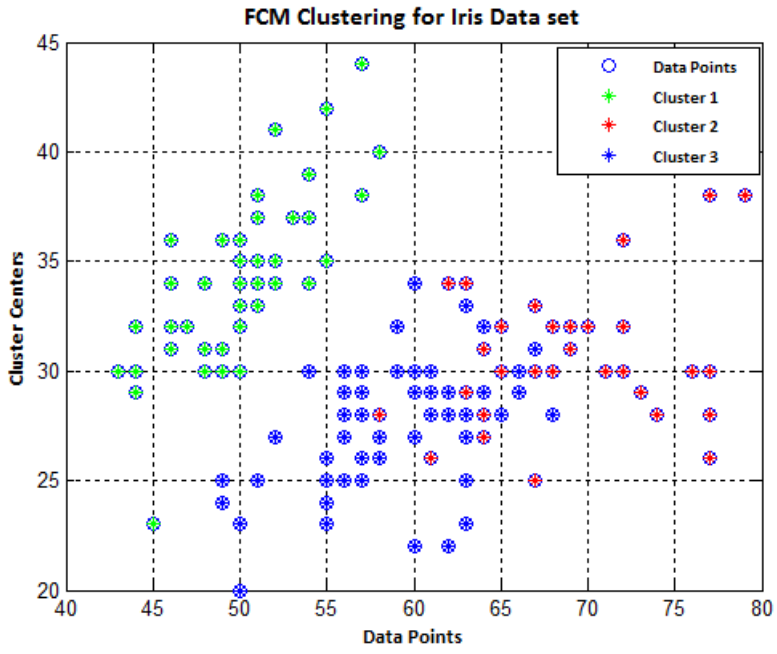


Figure 4.8: FCM Clustering for Iris Dataset

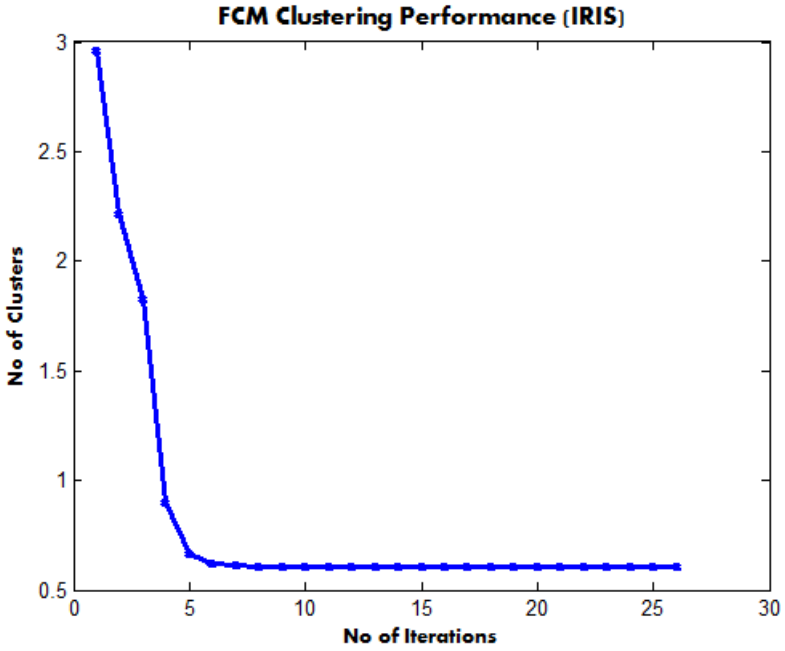


Figure 4.9: FCM Clustering Performance for Iris Dataset

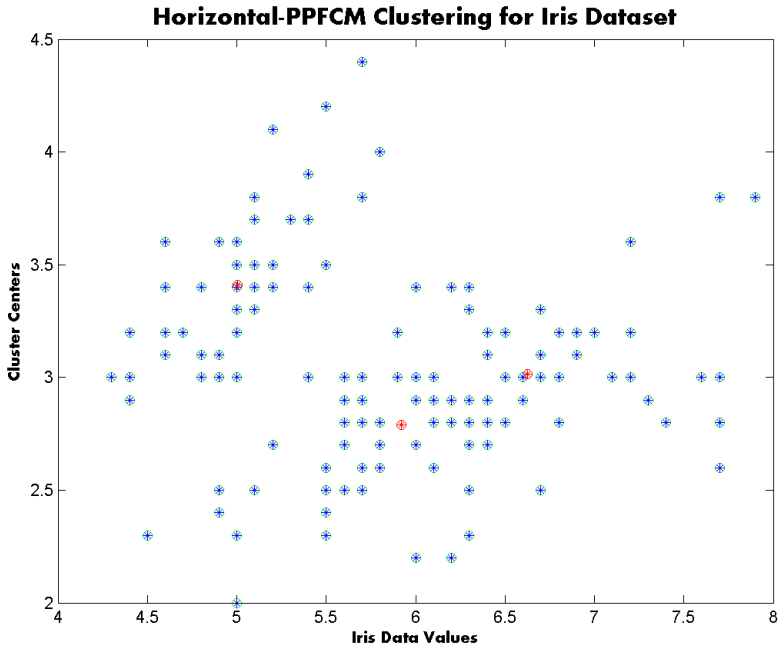


Figure 4.10: Horizontal-PPFCM Clustering for Iris Dataset

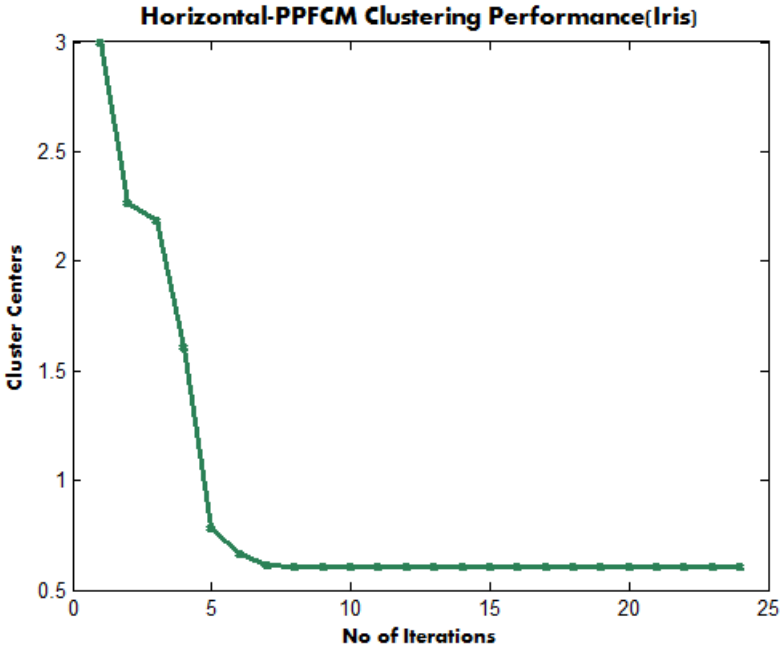


Figure 4.11: Horizontal-PPFCM Collaborative Clustering Performance for Iris Dataset

Glass Identification Dataset: Figures shows the non privacy preserving FCM clustering and its performance of glass identification dataset of 214 instances. Next figures gives the details of privacy preserving fuzzy c means clustering performed on glass identification dataset, when data is horizontally partitioned and distributed equal number of samples to two parties (107 for each party). It also shows collaboration function performance while finding the collaborative outcomes.

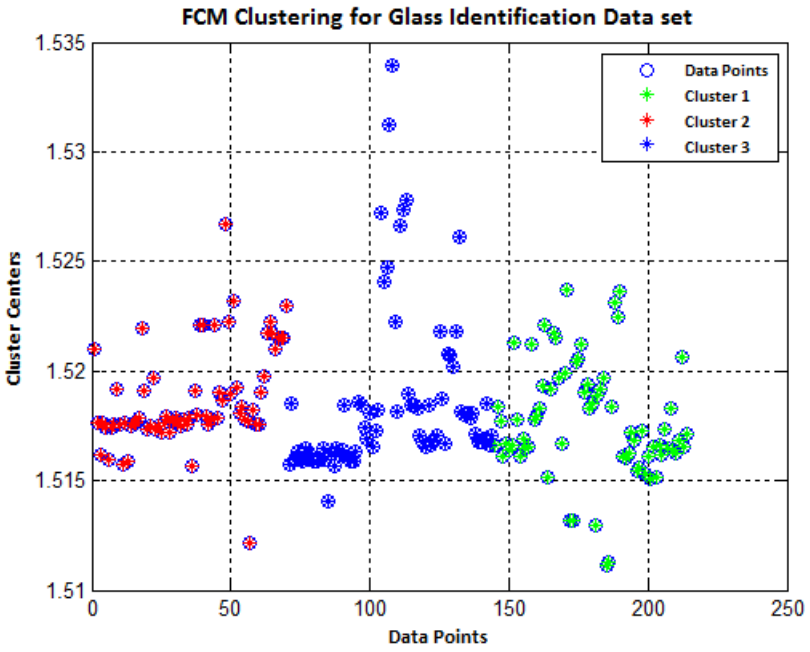


Figure 4.12: FCM clustering for Glass Identification Dataset

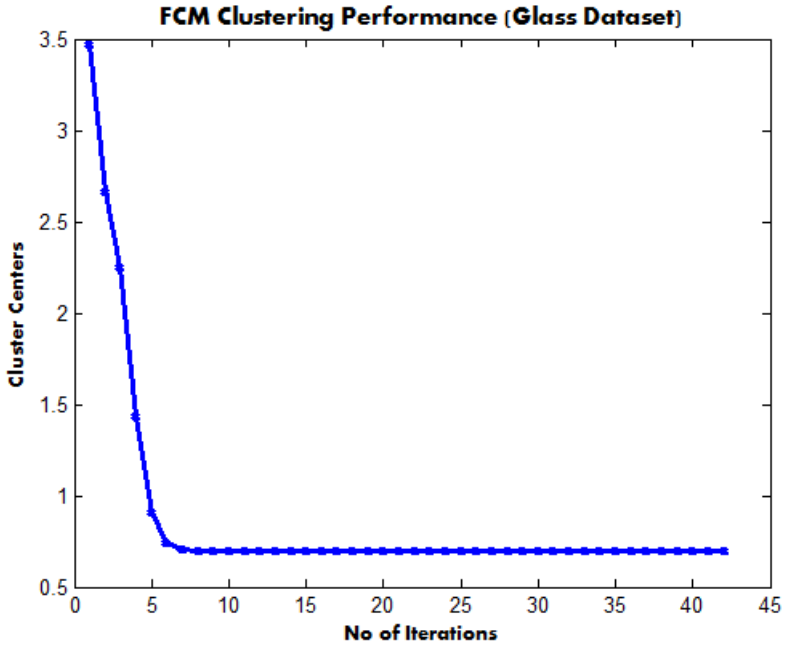


Figure 4.13: Horizontal-PPFCM Clustering for Glass Dataset

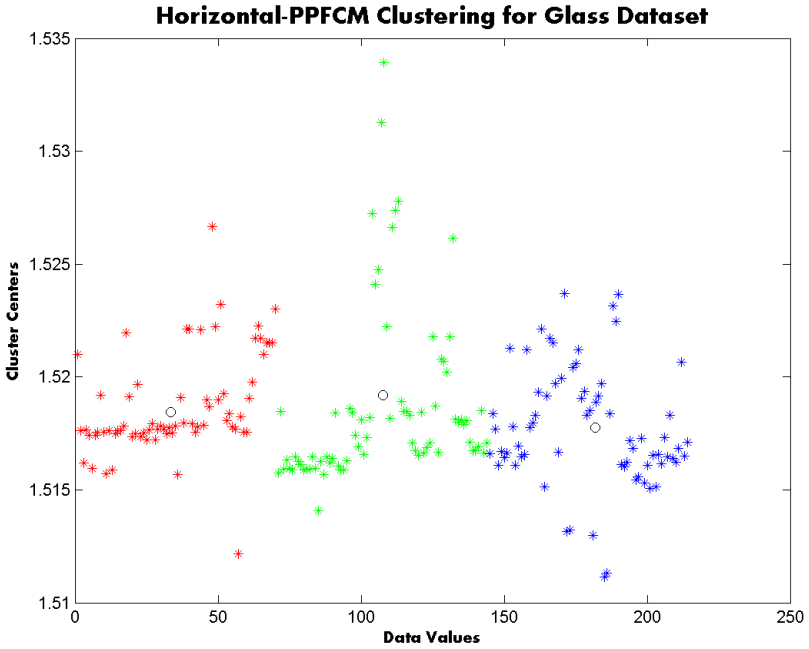


Figure 4.14: FCM clustering Performance of Glass Identification Dataset

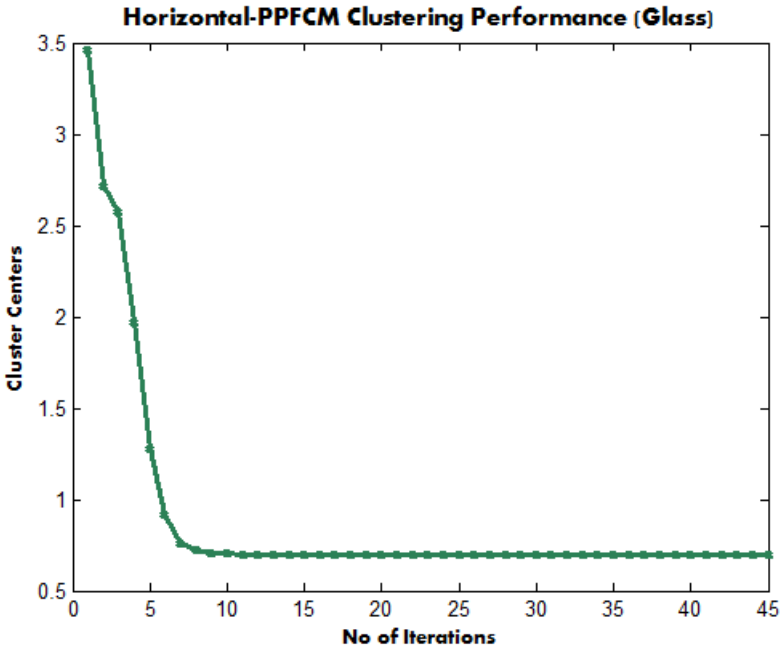


Figure 4.15: Horizontal-PPFCM Clustering Performance for Glass Dataset

Seeds Dataset: Clustering and performance of non privacy preserving FCM of Seeds dataset that has total 210 instances is shown in figures. Next results are presented for privacy preserving version of fuzzy c means clustering on seeds dataset. The data set is partitioned into two half with 105 samples for each partition has been given to two parties, then trained for privacy preserving collaborative clustering and results are shown for PPFCM clustering and its performance.

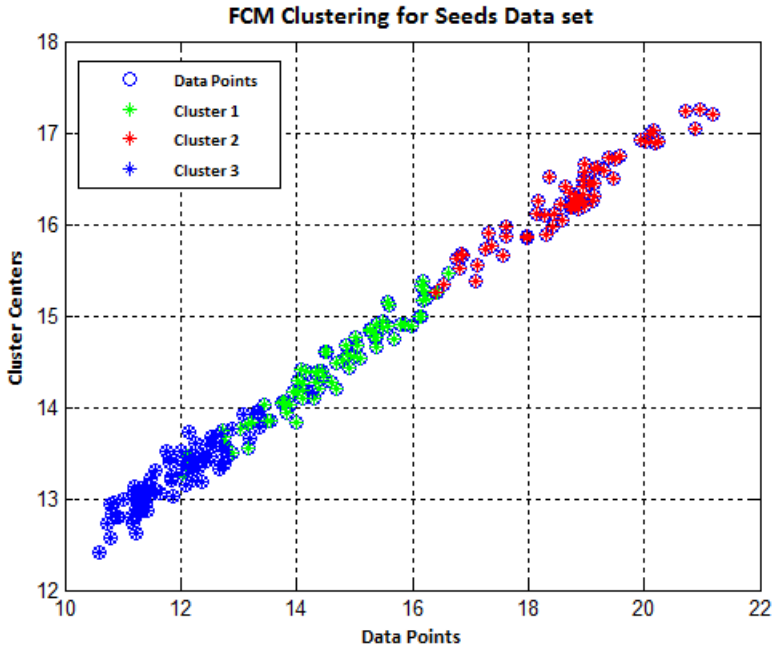


Figure 4.16: FCM clustering of Seeds Dataset

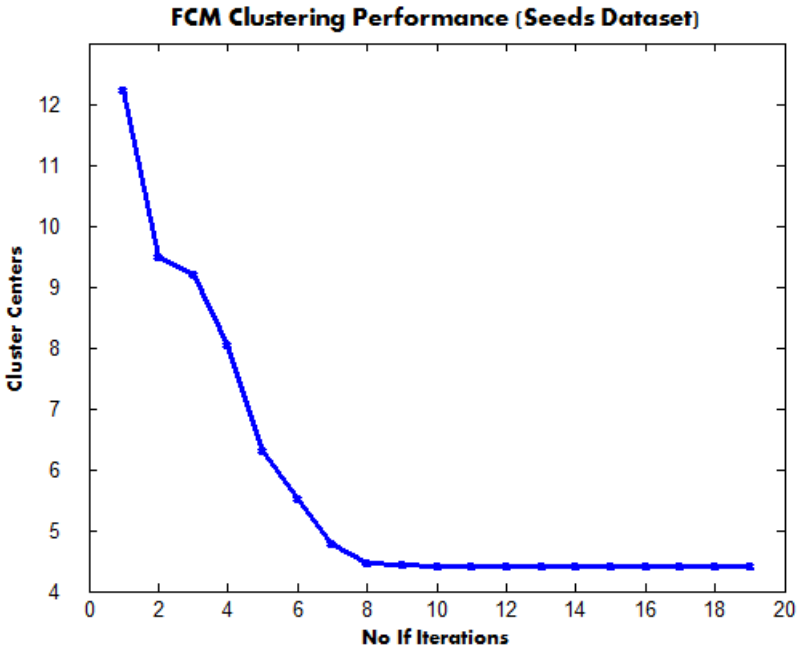


Figure 4.17: FCM clustering Performance of Seeds Dataset

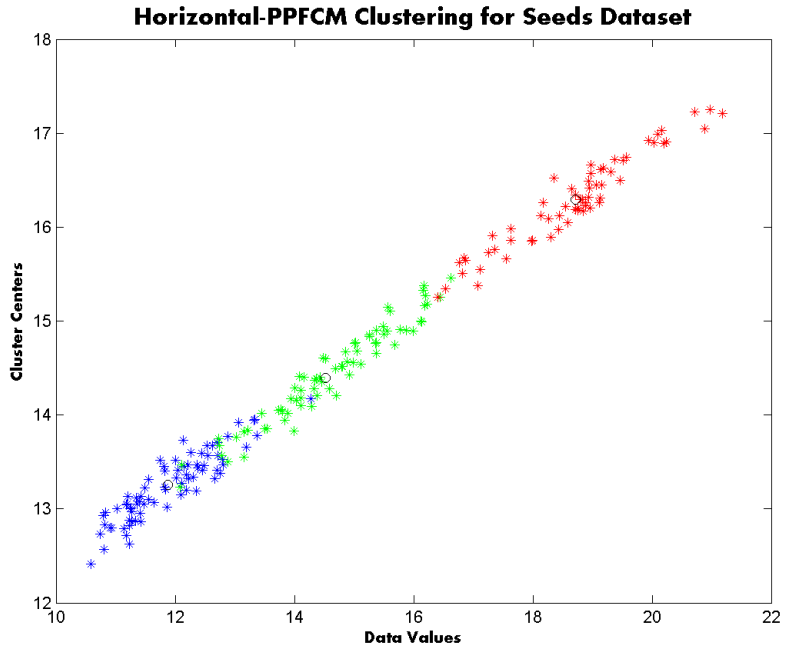


Figure 4.18: Horizontal-PPFCM Clustering for Seeds Dataset

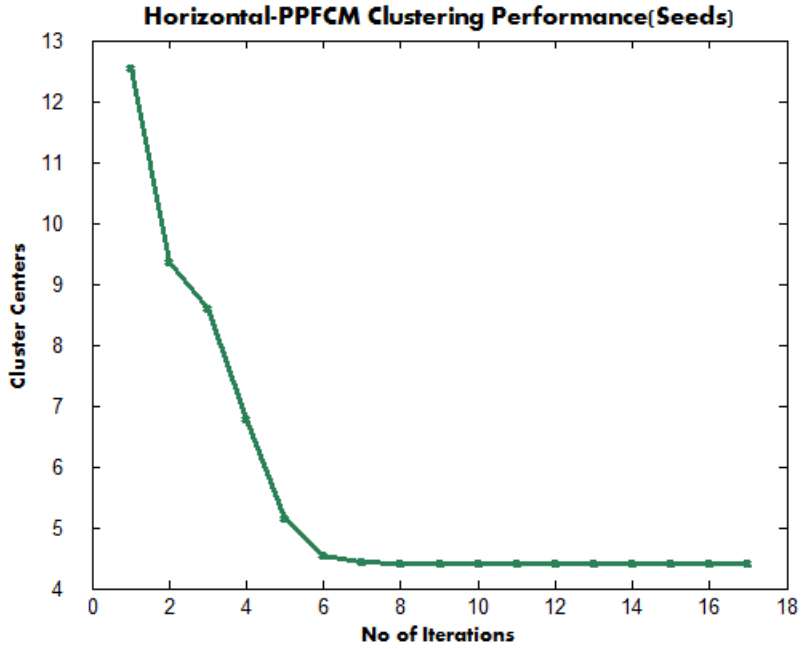


Figure 4.19: Horizontal-PPFCM Clustering Performance for Seeds Dataset

Wine Dataset: These results are shown for wine dataset which has 178 instances. The FCM clustering and performance of objective function of FCM clustering is presented on wine dataset. Next figures shows the collaborative fuzzy c means clustering of wine dataset, along with the performance of collaboration function of PPFCM, when dataset is horizontally partitioned.

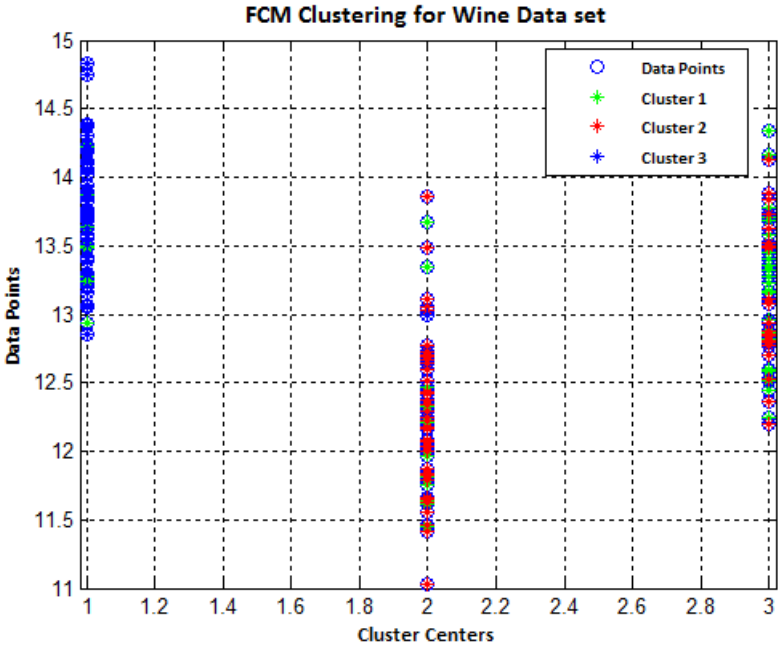


Figure 4.20: FCM Clustering for Wine Dataset

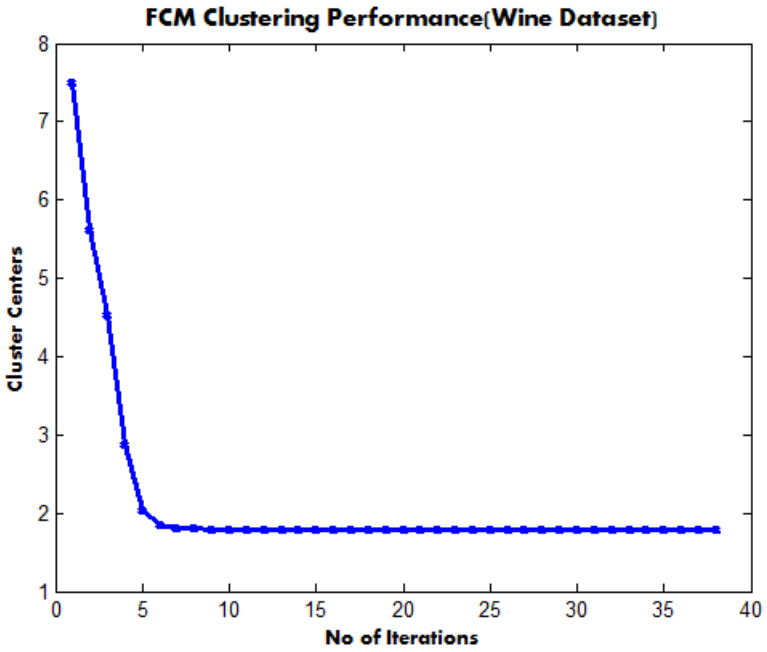


Figure 4.21: FCM Clustering Performance for Wine Dataset

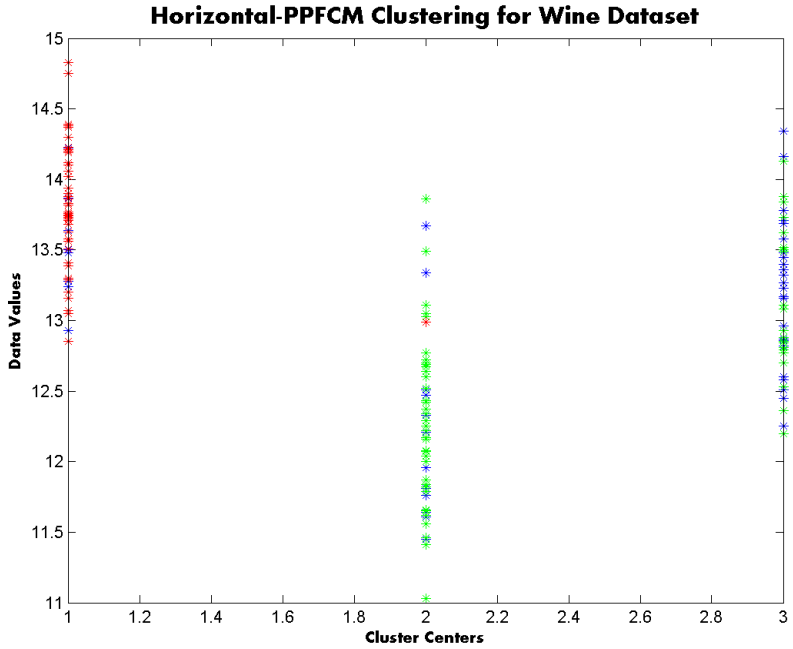


Figure 4.22: Horizontal-PPFCM Clustering for Wine Dataset

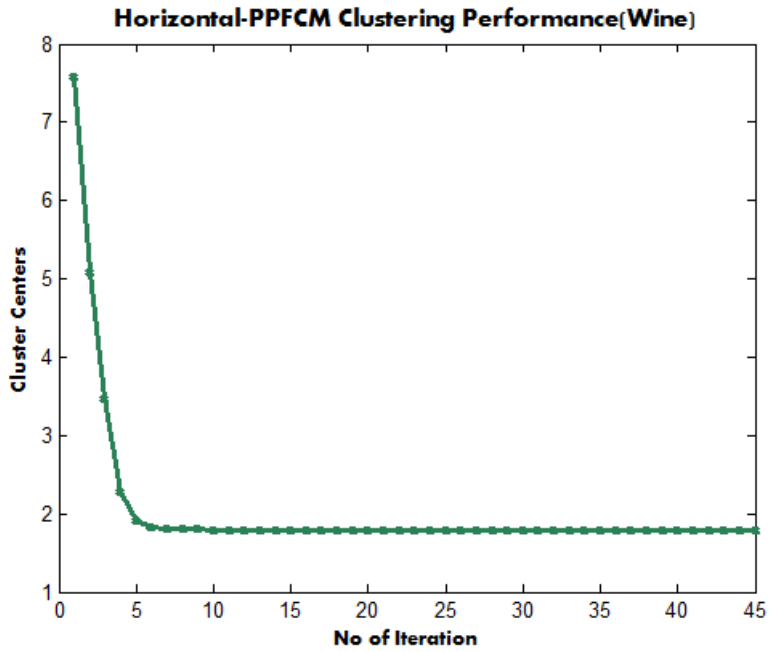


Figure 4.23: Horizontal-PPFCM Clustering performance for Wine Dataset

4.7.2 Cluster Centers in Horizontal-PPFCM Clustering

Prototypes of the individual data sites of individual parties before collaboration and after collaboration are presented in table below. In collaboration

Table 4.2: Cluster Centers in Collaboration of Horizontal-PPFCM

Dataset	**Cluster Centers in Horizontal-PPFCM**
Iris	0.6894 , 0.5824 , 0.3427
Glass	1.2510 , 4.2500 , 7.9774
Seeds	3.1884 , 3.1405 , 2.6950
Wine	0.7578 , 1.1637 , 2.3580

process, the prototypes (cluster centers) derived at individual parties are shown in this table. It is also considered as the collaboration table generated in the complete collaborative clustering process. The value of collaborative coefficient α=2.0 that impacts on the performance of collaboration function. The table shows the cluster centers reported for each dataset. The message will be shared between parties until there will not be any noticeable change in the pattern of collaboration. The results are feasible while preserving privacy in the fuzzy collaboration process.

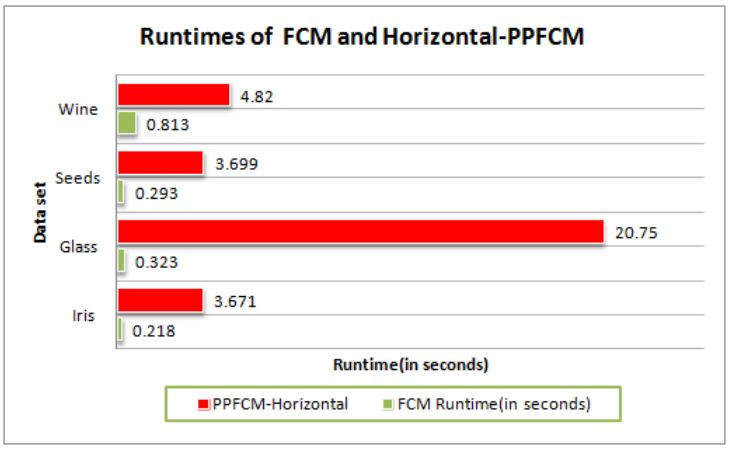
4.7.3 Time Complexity & Accuracy of Horizontal-PPFCM

Table shows the total execution time (Runtime) of FCM clustering and Horizontal-PPFCM before and after collaboration respectively, for four different datasets. The run time increased with number of times collaborations happened between parties.

Table 4.3: Runtimes of FCM and Horizontal-PPFCM

Runtime(in seconds)	**FCM**	**PPFCM-Horizontal**
Iris	0.218	3.671
Glass	0.323	20.75
Seeds	0.293	3.699
Wine	0.813	4.82

Figure 4.24: Runtime Graph for Horizontal-PPFCM

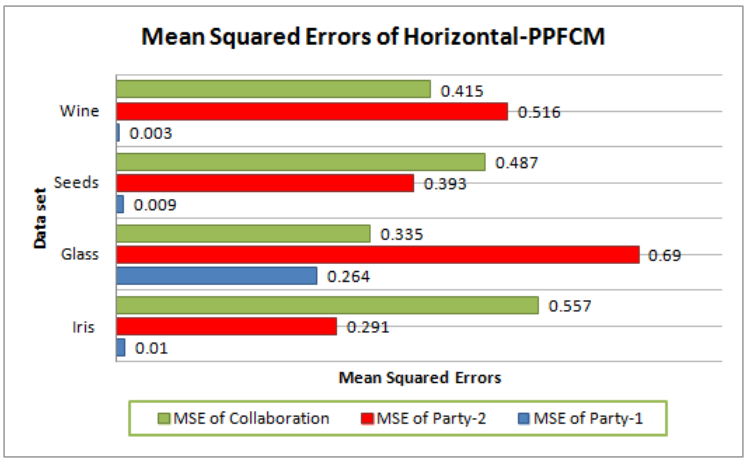


The accuracy of Horizontal-PPFCM with respect to the mean square errors in collaboration is presented in following table. Mean squared error is measured between two parties in collaboration process for 4 datasets.

Table 4.4: Mean Squared Errors of Horizontal-PPFCM

MSE Error	Party-1	Party-2	Collaboration
Iris	0.010	0.291	0.557
Glass	0.264	0.690	0.335
Seeds	0.009	0.393	0.487
Wine	0.003	0.516	0.415

Figure 4.25: Mean Squared Errors of Horizontal-PPFCM



4.7.4 Privacy Analysis of Horizontal-PPFCM

Table 4.5: Privacy Metrics Results and Analysis of Horizontal-PPFCM

Privacy Metric	Range & Level	Metric Result	Privacy Level
priv_CS	[0, 1] ($\leq$0.5=Low)	0.25	**High**
priv_TC	[0, ∞] ($\leq \infty$=Low)	1.11	**High**
priv_PS	[0, ∞] ($\leq \infty$=Low)	1.00	**High**

- ***Privacy of Granular Information:*** The granular information means the input level outcomes to share with other party in collaboration process. In PPFCM method the inputs are shared in the form of granular information of the input data. The cluster centers and the partition matrices are shared between parties instead if direct input attribute values. The fuzzy partitions and their cluster centers does not carry any original or sensitive information to other party, hence the privacy is well preserved in this phase. The privacy measured at this stage using cluster similarity $priv\_CS$ where the similarity between the clusters formed by the parties are compared and if the difference is Low then the privacy is said to be high. The privacy level of Horizontal-PPFCM has shown high level privacy in this input phase.

- ***Privacy of Parties in Collaboration:*** The main phase of collaboration process, where privacy of parties to be highly preserved is the collaboration phase. Collaborating parties expect high privacy level in collaboration phase as they share the information by exchanging prototypes to each other. The privacy metric t-Closeness $priv\_TC$ is used to evaluate the privacy level of parties, where the distribution of original input values must be close to the distribution of the shared information in collaboration. The difference (distance) between two parties must be small to gain high privacy. The result shows low value and the high privacy level for Horizontal-PPFCM in this phase.

- ***Privacy of Outputs in Collaboration:*** The output level is the final phase where the combined results of the collaboration are published. The collective output from any collaborative process must ensure privacy of parties involved. Privacy score $priv\_PS$ measures the privacy level assured in collaboration. Privacy score indicates the privacy risk increases with the sensitivity of information granules and their visibility in collaboration. Low visibility decreases the privacy risk and gives

high privacy level. Horizontal-PPFCM shows the high privacy level by showing low visibility of information in collaboration

4.7.5 Results of Vertical PPFCM

All the experiments are carried out for Iris,Glass,Seeds & Wine datasets. Vertical-PPFCM algorithm has shown following results with collaboration coefficient $\beta = 2.0$ for both parties. The results are derived before and after collaboration, and the impact of party 1 on party 2 and impact of party 2 on party 1 in the collaboration process.

Iris Dtaset: When the iris dataset is vertically partitioned, each party holds 150 instances but only two attribute values resides at each party (sepal length, sepal width at party-1 and petal length, petal width at party-2) class labels Setosa, Versicolour, Virginica are known two both parties. Figures show the collaborative clustering and the performance of collaboration function.

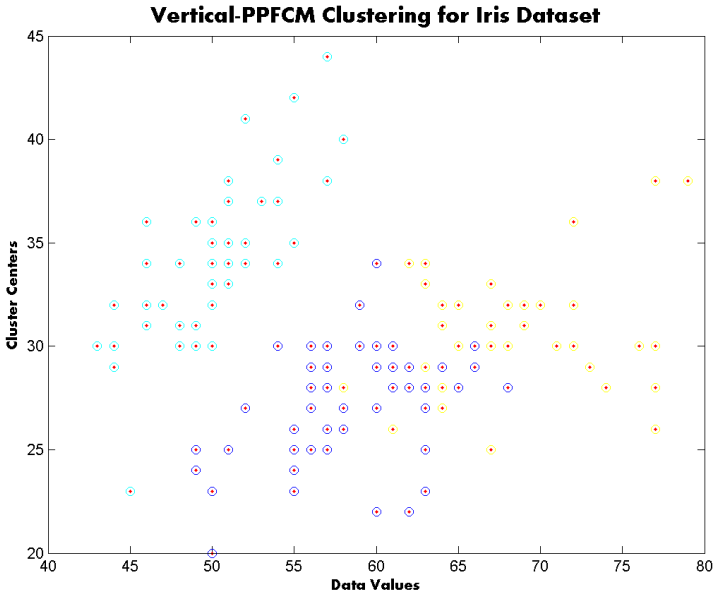


Figure 4.26: Vertical-PPFCM Clustering for Iris Dataset

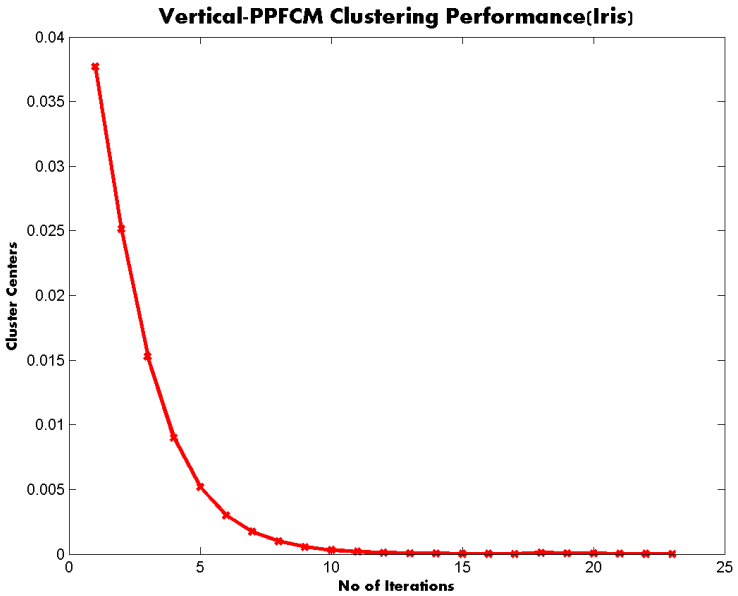


Figure 4.27: Vertical-PPFCM Clustering Performance for Iris Dataset

Glass identification Data set: When glass identification data set with 214 instances, is vertically partitioned and distributed to both parties (Id number, RI, Na, Mg, Al at party-1) and (Si, K, Ca, Ba, Fe are at party-2). Type of glass is known to both parties. The collaborative clustering results are shown in figures.

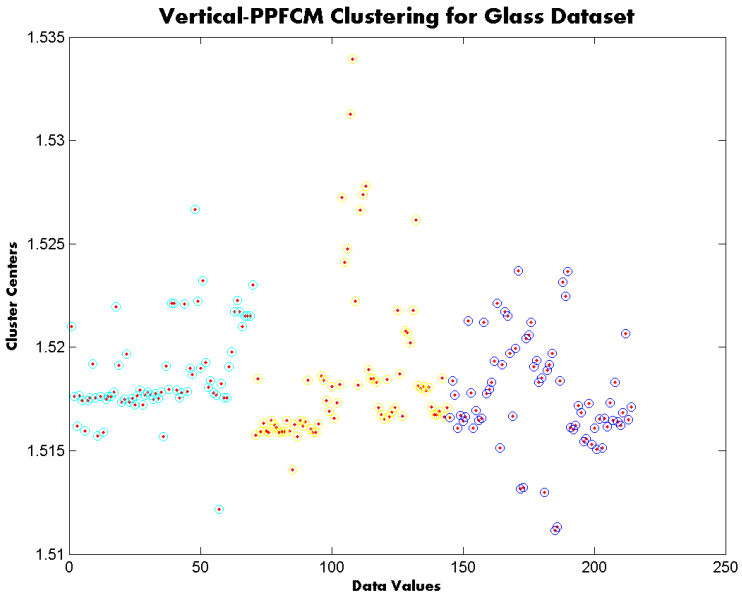


Figure 4.28: Vertical Collaborative FCM Clustering for Glass Dataset

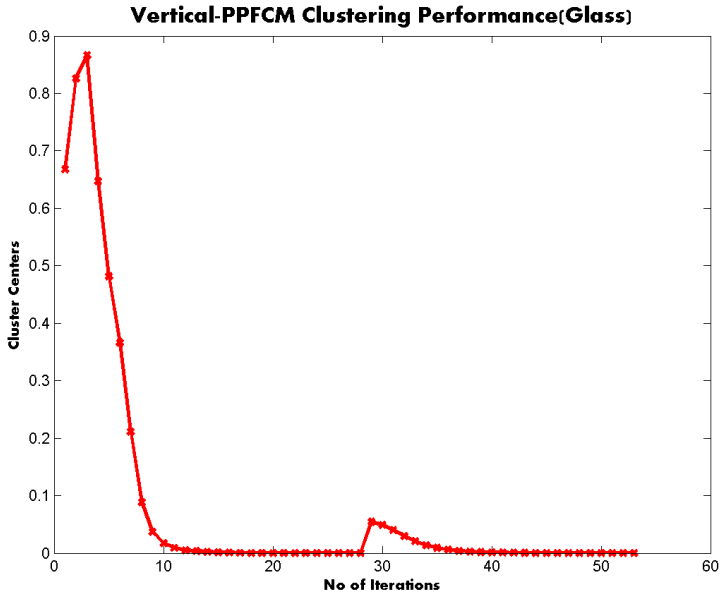


Figure 4.29: Vertical-PPFCM Clustering for Glass Dataset

Seeds Dataset: Seeds data set that has total 210 instances is vertically partitioned into two partitions where party 1 holds area A, perimeter P,

compactness $C = (4 * \pi * A/P^2)$ and party 2 holds length of kernel, width of kernel, asymmetry coefficient. Class attribute length of kernel groove is known to both the parties. results are shown in figures.

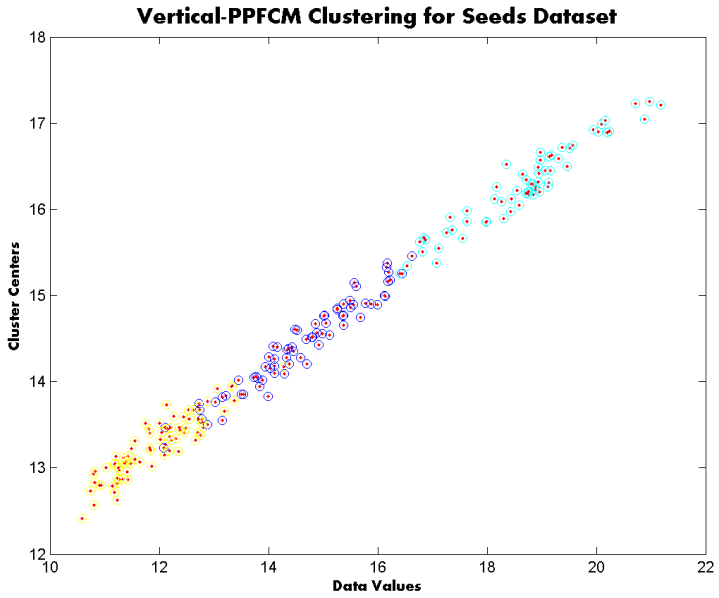


Figure 4.30: Vertical Collaborative FCM Clustering for Glass Dataset

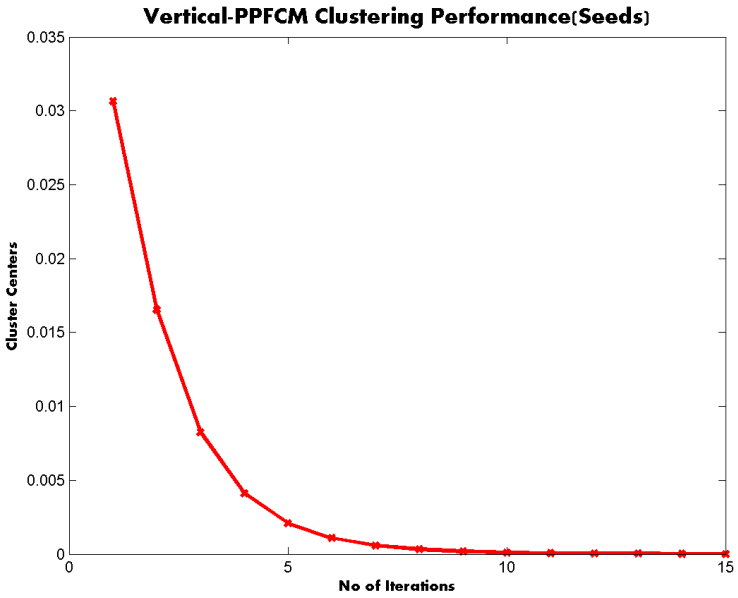


Figure 4.31: Vertical-PPFCM Clustering for Seeds Dataset

Wine Dataset: The data set with 178 instances is vertically partitioned and distributed where party-1 holds Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols and party-2 holds Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines. Class attribute Proline is known to both parties and results are shown in figures.

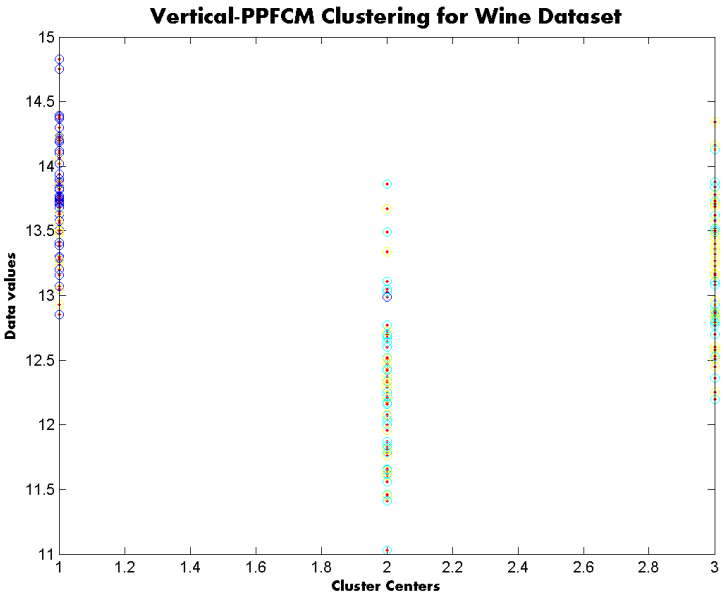


Figure 4.32: Fuzzy C-Means Clustering for Wine Dataset

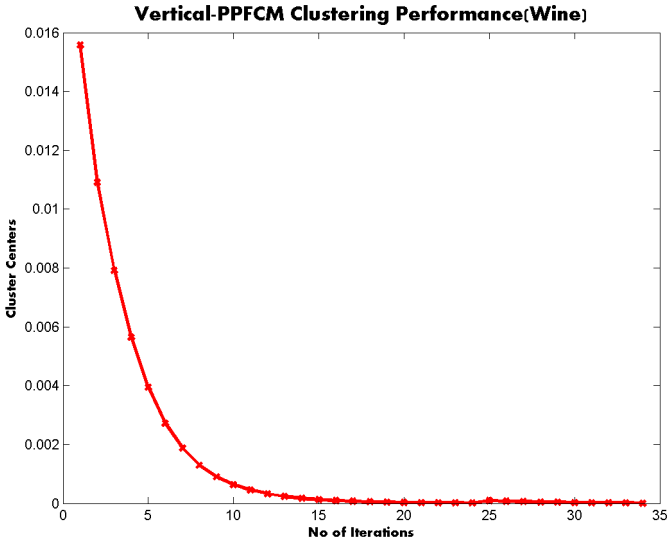


Figure 4.33: Vertical-PPFCM Clustering and Cluster Centers for Wine Dataset

4.7.6 Cluster Centers in Vertical-PPFCM

Prototypes that are produced in vertical collaboration process are given in table below. Prototypes are the cluster centers generated and exchanged at the time of collaboration between parties.

Table 4.6: Cluster Centers in Process of Vertical-PPFCM

Dataset	**Cluster Centers in Vertical-PPFCM**
Iris	5.922 , 2.7889 , 4.3971
Glass	1.5177 , 13.1935 , 3.2859
Seeds	14.4078 , 5.5077 , 2.8104
Wine	20.7774 , 92.3937 , 2.0683

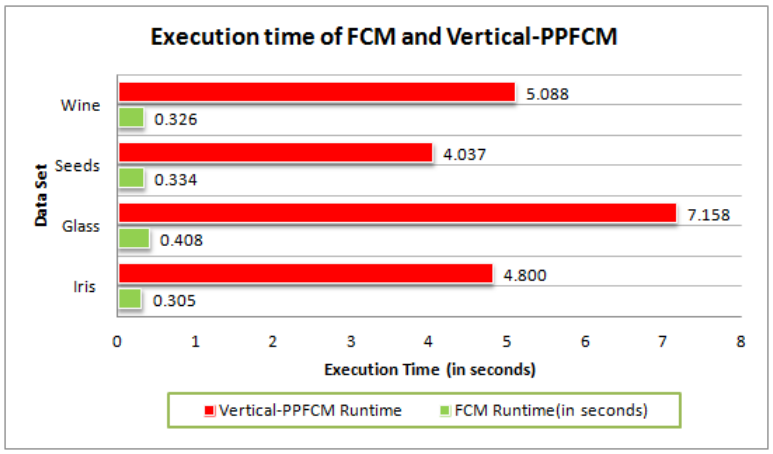
4.7.7 Time Complexity & Accuracy of Vertical-PPFCM

The execution time of the Vertical-PPFCM algorithm is given for each dataset, compared with non collaboration algorithm. Table 4.7 shows the total execution time (run time) of FCM clustering and Vertical-PPFCM before and after collaboration respectively.

Table 4.7: Execution time of FCM and Vertical-PPFCM

Runtime(in seconds)	**FCM**	**Vertical-PPFCM**
Iris	0.305	4.800
Glass	0.408	7.158
Seeds	0.334	4.037
Wine	0.326	5.088

Figure 4.34: Runtime Graph for Vertical-PPFCM

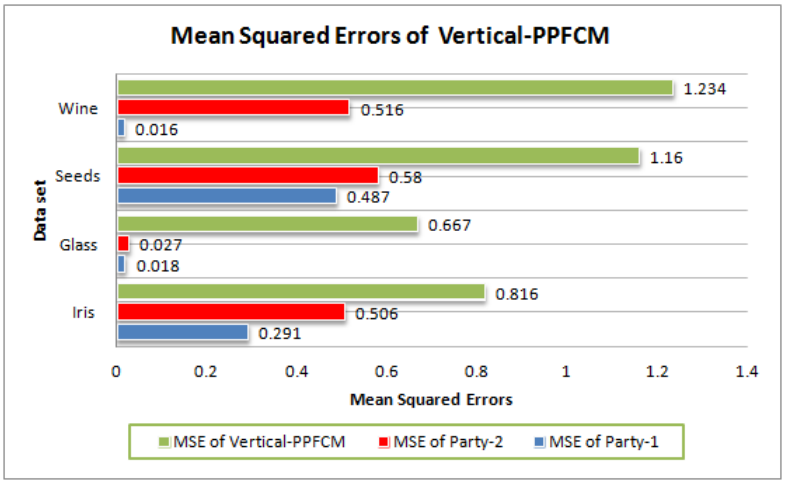


A table of mean squared errors of parties and their collaboration in vertical PPFCM algorithm for all datasets is given below

Table 4.8: Mean Squared Errors of Collaborating Parties in Vertical-PPFCM

MSE Error	Party-1	Party-2	Collaboration
Iris	0.291	0.506	0.816
Glass	0.018	0.027	0.667
Seeds	0.487	0.580	1.160
Wine	0.016	0.516	1.234

Figure 4.35: Mean Squared Errors of Vertical-PPFCM



4.7.8 Privacy Analysis of Vertical-PPFCM

Table 4.9: Privacy Metrics Results and Analysis of Vertical-PPFCM

Privacy Metric	Range & Level	Metric Result	Privacy Level
priv_CS	[0, 1] ($\leq$0.5=Low)	0.18	**Low**
priv_TC	[0, ∞] ($\leq \infty$=Low)	2.45	**High**
priv_PS	[0, ∞] ($\leq \infty$=Low)	0.01	**High**

- ***Privacy of Granular Information:*** The input level outcomes in the form of granular information are shared with other parties in collaboration process. The cluster centers and the partition matrices are shared between parties instead of direct input values. The privacy measured at this stage using cluster similarity $priv\_CS$ where the similarity between the clusters formed by the parties are compared and if the difference is Low then the privacy is said to be high. The privacy level of Vertical-PPFCM has shown high level privacy in this input phase.
- ***Privacy of Parties in Collaboration:*** Vertical collaborating parties expect high privacy level in collaboration phase as they share their information (partial) through prototypes to each other. The privacy metric t-Closeness $priv\_TC$ is used to evaluate the privacy level of parties, where the distribution of original input values must be close to the distribution of the shared information in collaboration. The difference (distance) between two parties must be small to gain high privacy. The result shows low value and the high privacy level for Vertical-PPFCM in this phase.
- ***Privacy of Outputs in Collaboration:*** The collective output from any collaborative process must ensure privacy of parties involved. Privacy score $priv\_PS$ measures the privacy level assured in collaboration. Privacy score indicates the privacy risk increases with the sensitivity of information granules and their visibility in collaboration. Low visibility decreases the privacy risk and gives high privacy level. Vertical-PPFCM shows the high privacy level by showing low visibility of information in collaboration

4.7.9 Complexity analysis & Scalability of PPFCM:

In both Horizontal and Vertical PPFCM, datasets used are with instances up to 300 and computational complexity has shown high when number of

parties increased and number of secure computations are done in every level of communication. In Horizontal-PPFCM sequential collaboration is used for privacy preserving, hence the computational complexity increases for every level. In Vertical PPSOM central collaboration is used hence the computational overhead is increased. When PPFCM methods tested for a larger dataset like cloud data with 2053 instances and 10 attributes, it could not score acceptable accuracy and privacy level.

4.8 Chapter Summary

The Chapter named Privacy Preserving Fuzzy C-Means Clustering, given the complete information of Fuzzy C-Means Clustering of distributed data. This chapter described ways of collaboration for preserving privacy of parties while exchanging the intermediate results with each other. The collaborative clustering process has been undertaken and implemented for horizontal & vertical data distribution mechanisms for two parties. Experiments and Results of the proposed algorithms along with the privacy analysis of both Horizontal-PPFCM and Vertical-PPFCM are presented.

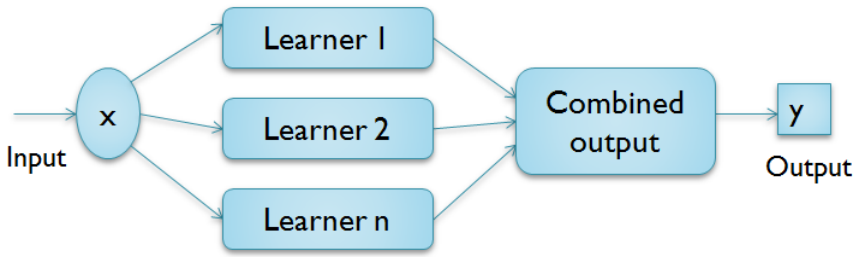
Chapter 5

Privacy Preserving Global Random Forest Classification

5.1 Ensemble Learning

Ensemble Learning method has the capability of learning from multiple classifier systems. This learning mechanism allows multiple base learners applied for a training data set through multiple base learning algorithms to solve same problem. It constructs a set of learners and combine them. The generalization ability of an ensemble is much stronger than that of base learners. Ensemble methods are able to boost learning ability and performance of weak learners (base learners)

Figure 5.1: Ensemble Learning Network

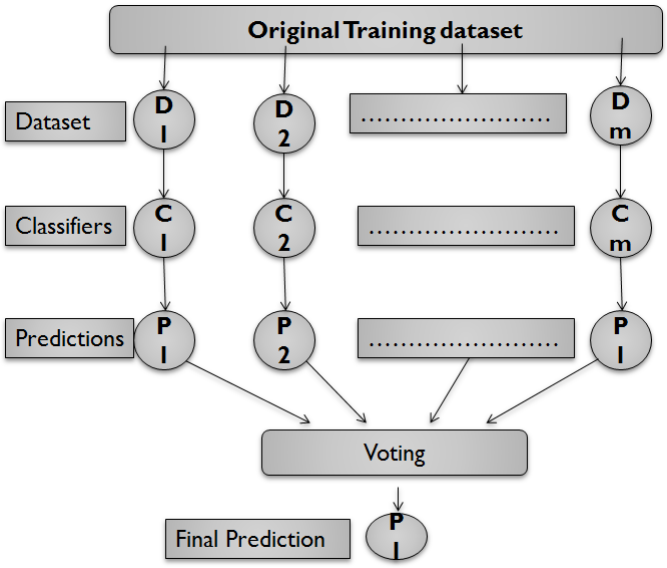


5.1.1 Bootstrap Aggregating - Bagging

The Bagging is a bootstrap aggregating learning method, that has the ability of improving stability and accuracy of a learning algorithm. The Bagging is an efficient method of bootstrapping of machine learning algorithms that is

mostly used for improved classification and regression models. The bagging is an ensemble learning method that also helps in reducing statical variance and also avoid over fitting. This meta-ensemble learning method usually applied in decision tree classification to improve it's learning and classification accuracy.

Figure 5.2: Bagging Ensemble Learning Process



An ensemble method establishes multiple ensembles by training individual network of each learner over random redistribution of original training data set. In this method some of the patterns may be repeated or duplicated in ensemble learning process and not all the patterns of original training set are included in a member ensemble.

5.1.2 Random Forest Classification

A Random forest is a classification method where multiple number of decision trees are combined together as an ensemble in order to get improved stability and accuracy prediction results. Generally a random forest is known as an ensemble of randomly chosen decision trees based on majority of classes through voting. The main objective of any ensemble learning method is combine learning models aimed for boosting final results. The major advantage of a random forest classification is that the model can be used for both classification and regression problems[37]. Random forest is a supervised learning

method and each decision tree becomes a building block. A random forest is built by multiple decision trees over data samples and get prediction results from each decision tree, then produce best solutions after voting process. In general decision trees have two main properties that are low bias and high variance[5]. When the decision tree is created with too much depth, then it is called over fitting that results in high error in test data.

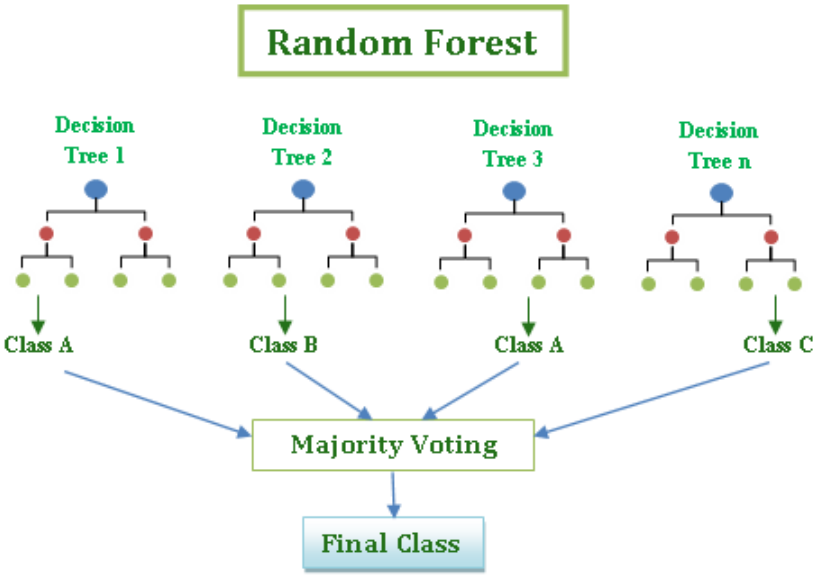


Figure 5.3: Random Forest

Why Random Forest? In Random forest decision trees are more independent because of random feature selection where as in bagging all the features are selected for splitting a single node. Hence a random forest can produce a better prediction results in less time as they earn from only subset of features. The Random forest classification algorithm is given below.

Algorithm 8 Random Forest Classification Algorithm

Input: Records R, attributes A, class attribute c, randomization parameter s, tree depth δ, number of trees o.
Output: An Ensemble of o decision trees.
For $k \leftarrow 1$ to o Do
$R \leftarrow$ randomly select n=$|R|$ records out of R with replacement
$tree^k \leftarrow$ Recursive_Random_Tree (R,A,c,s,δ)
End
return $\{tree^1, tree^2, ...tree^o\}$

5.2 Privacy Preserving Global Random Forest Classification-Horizontal

Privacy preserving Random Forest classification adopts horizontal partitioning to preserve privacy while constructing number of decision trees in Random forest. The data is horizontally partitioned and distributed to the parties to be joined in random forest classification. The proposed algorithm uses C4.5 algorithm repeatedly and generates decision trees form feature subsets and aggregates them to build a final random forest without violating privacy. Privacy preserving algorithm of building a global random forest from all the local random forests is presented here.

Algorithm 9 Privacy Preserving Global Random Forest Classification-Horizontal

---

Partitioning: Dataset D_n, partitioned horizontally
Initialize: $D(P_1) = \{d_i, d_i + 1, ...d_s\}$, $D(P_2) = \{d_j, d_j + 1....k\}$.
Input: Records R, attributes A, class attribute c, randomization parameter s, tree depth δ, number of trees o.
begin:
Step 1: ***Local Decision Trees:***
Party P_1 builds decision trees $t_i = DTC4.5(d_i)$ For each $d_i \in (d_i, ...d_s)$.
Party P_2 builds decision trees $t_j = DTC4.5(d_j)$ For each $d_j \in (d_j, ...d_k)$.
Step 2: ***Local Random forests:***
Party 1 builds $LRF_1 = Unique(t_i)$
Party 2 builds $LRF_2 = Unique(t_j)$
Step 3: ***Aggregation:*** Initial Global Random Forest-1
$IGRF_1 = Unique(LRF_1, LRF_2)$.
Step 4: ***Local Voting:*** on left out trees in P_1 and P_2
Party 1 builds $LRF_{1A} = Majclass(t_i)$
Party 2 builds $LRF_{2B} = Majclass(t_j)$
Step 5: ***Append:*** Initial Global Random Forest-2
$IGRF_2 = Mejclass(LRF_{1A}, LRF_{2B})$.
Step 6: ***Final global random forest:***
$HFGRF = AGGR(IGRF_1, IGRF_2)$
End

Here Privacy preserving is achieved by hiding original datasets at local level, and with the secured way of sending only the classified data of the local parties to the aggregation[3]. Each party performs independent voting at their local levels in the process of building a final global random forest. In this entire process, the only information that is commonly shared or known

to all the parties is Initial and Global Random forest and the final global random forest. Hence the objective of preserving privacy in random tree classification has been successfully achieved.

5.2.1 Process of Horizontal-PPGRF Algorithm

The complete process of privacy preserving random forest classification is presented in process flow diagram below.

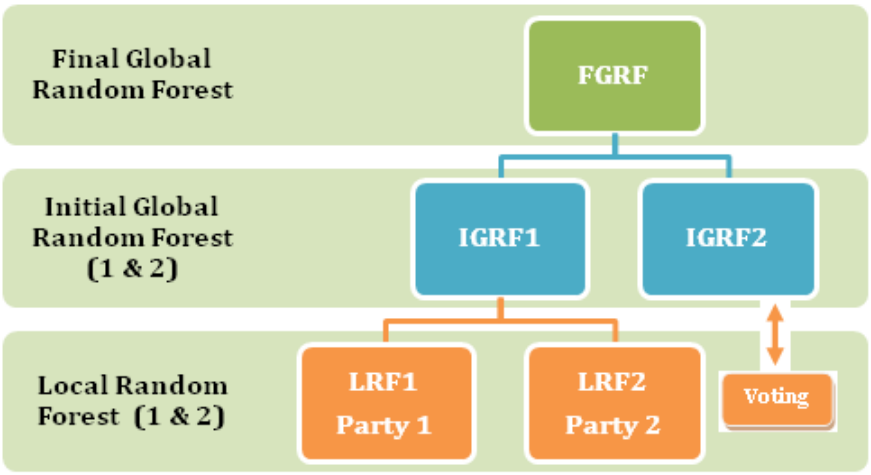


Figure 5.4: The Process flow of Privacy Preserving Global Random Forest classification-Horizontal

1. *Local Random Forest Construction:* For each party, subsets of their own dataset are selected randomly to generate decision trees and based on these subsets a local Random Forest is generated by the local parties at their own sites, then sends to the Ensemble Aggregation

2. *Secured Ensemble Aggregation:* The ensemble aggregation securely combine all the ensembles received from local parties and also removes the redundancies then builds an initial global random forest.

3. *Local Voting:* Local vote manager receives the initial global random forest and specifies or vote for the majority of class at their site and sends the updated initial global random forest back to the other parties.

4. *Final Global Random Forest:* After receiving votes from all the parties, the last party receives the voted initial global random forest and joins all the voted trees of local random forests to form final ensemble of decision trees i.e., the Final Global Random Forest.

5.3 Privacy Preserving Global Random Forest Classification-Vertical

Privacy preserving Global Random Forest classification for vertical data distribution adopts three levels of privacy preserving, (a). to preserve privacy while constructing number of decision trees in Random forest, (b) aggregating the local random forests of local decision trees, and (c) Voting on local random forests for the final aggregation. Algorithm proposed for Random Forest Classification for privacy preserving of vertically distributed data is given below.

Algorithm 10 Privacy Preserving Global Random Forest Classification-Vertical

---

Vertical Partitioning: of given dataset $D(n) = \{d_i, d_i+1, d_j, d_j+1, ...d_n\}$,
Party-1 holds $D(P_1) = \{d_i, ...d_n\}$, attribute set $(A_i) = \{a_i, a_i+1, ...a_j\}$
Party-2 holds $D(P_2) = \{d_i, ...d_n\}$ attribute set $(A_j) = \{a_j, a_j+1....k\}$
Input: Records R, attributes A, class attribute c, randomization parameter s, tree depth δ, number of trees o.
Begin: For Parties P_1&P_2
Step 1: ***Local Decision Trees:***
Party P_1 builds decision trees $t_i = DTC4.5(d_i)$ For each $d_i \in (d_i, ...d_s)$.
Party P_2 builds decision trees $t_j = DTC4.5(d_j)$ For each $d_j \in (d_j, ...d_k)$.
Step 2: ***Local Random forests:***
Party 1 builds $LRF_1 = Unique(t_i)$
Party 2 builds $LRF_2 = Unique(t_j)$
Step 3: ***Aggregation:*** Initial Global Random Forest-1
$IGRF_1 = Unique(LRF_1, LRF_2)$.
Step 4: ***Local Voting:*** on unique trees in LRF_1 and LRF_2 of P_1 and P_2
Party 1 builds $LRF_{1A} = Unique(Majclass(t_i))$
Party 2 builds $LRF_{2B} = Uique(Majclass(t_j))$
Step 5: ***Append:*** Initial Global Random Forest-2
$IGRF_2 = Unique(LRF_{1A}, LRF_{2B})$.
Step 6: ***Final Global Random Forest:***
$VFGRF = AGGR(IGRF_1, IGRF_2)$
End

The algorithm recursively executes a C4.5 algorithm to build local decision trees and combine them into a local random forest by selecting unique trees. Party 1 builds LRF-1 and party 2 builds LRF-2, then after ensemble aggregation the privacy preserving algorithm builds an initial global random forest from LRF-1 and LRF-2 of local parties. In final stage the voting is

done at local parties based on the majority of class from classified trees of party 1 and party 2. A final global random forest (FGRF-Vertical) will be constructed.

When the dataset is vertically partitioned and distributed to the parties party 1 and 2, each party will have only few attribute values but all the samples at that site while performing random forest classification. Hence it is very important to securely aggregate the decision trees of the vertical partitions in-order to get an output which can have the complete results of both the parties. In entire process there in no violation of privacy happens as the algorithm preserves the privacy of local parties.

5.3.1 Process of Vertical-PPGRF Classification

The complete process of privacy preserving random forest classification is presented in process flow diagram, followed by steps involved for building privacy preserving Global Random Forest, when data is vertically distributed. The digram shows how patries independantly builds local random forests and how ensemble aggregation is performed to build IGRF-1, IGRF-2 and Final Global Random Forest, using Vertical-PPGRF method.

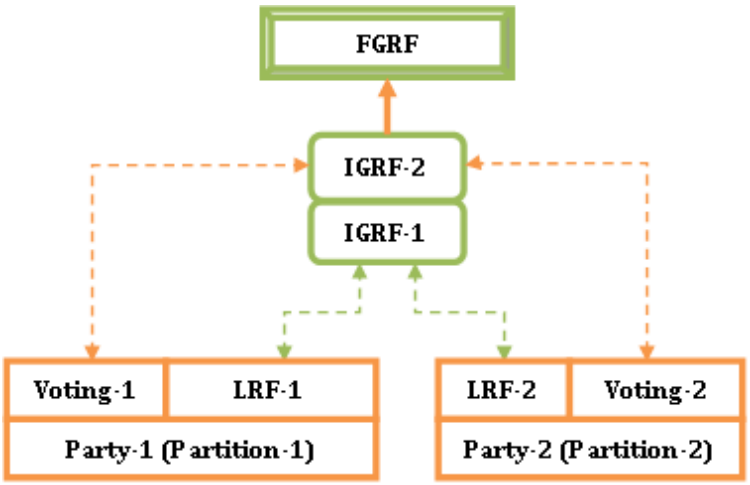


Figure 5.5: Process Flow of Privacy Preserving Global random forest Classification-Vertical

1. Distributing the data vertically between multiple parties.
2. Building Local Random Forests (LRF-1 & LRF-2) by parties 1 and 2.
3. Aggregating the local random forests using ensemble aggregation.

4. Building the Initial Global Random Forest-1
5. Voting on classified trees and building new LRF-1A and LRF-2B with majority class.
6. Aggregating LRF-1A and LRF-2B and building the Initial Global Random forest-2
7. Finally aggregation of IGRF-1 and IGRF-2 will be performed and Final Global Random Forest (V-FGRF) is constructed.

5.4 Experiments & Results

The experiments are carried out for iris, glass, seeds, and clinical datasets in python (anaconda environment). The results are produced for Horizontal-PPGRF and Vertical-PPGRF classification methods. In process of building a final global random forest there are three main privacy preserving phases to follow to build IGRF-1, IGRF-2 and FGRF for each dataset. each party generates specified number of decision tress in random forest (between 10 to 100 threshold), limiting tree depth to 5 (this can be varied for larger datasets). The attribute selection in decision trees is done by measuring entropy values.

5.4.1 Results of Horizontal-PPGRF

In First Phase, the process starts with partitioning the dataset into two horizontal partitions and will be distributed to two parties (p1 and p2). Then for each partition, generates specified number of decision tress in random forest. Then from generated decision trees, all unique decision trees are picked to form local random forests for two parties (Party1 builds LRF1 and Party2 builds LRF2). Now aggregating all the uniquely classified decisions trees in LRF-1 and LRF-2 to build an Initial global random forest-1 (IGRF-1). In next phase of computations local parties performs voting on miss classified trees and generates IGRF-2 by appending trees with majority class. The results are given in following tables for four datasets Iris, Seeds, Clinical and Seeds.

Now according the proposed algorithm a final global random forest is to be constructed by performing voting on initial global random forests (IGRF-1 & IGRF-2) generated by both the parties. In last phase, the final aggregation of IGRF-1 and IGRF-2 takes place to build The final global random forest(FGRF).

Table 5.1: IGRF-1 and IGRF-2 in Horizontal-PPGRF for Iris Dataset

Trees	LRF-1	LRF-2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	8	10	**18**	1	0	**1**
20	11	20	**31**	4	0	**4**
30	13	30	**43**	4	0	**4**
40	13	39	**52**	6	1	**7**
50	17	45	**62**	7	5	**12**
60	18	59	**77**	9	1	**10**
70	14	36	**50**	4	12	**16**
80	22	59	**81**	10	7	**17**
90	17	73	**90**	5	10	**15**
100	22	81	**103**	9	9	**18**

Table 5.2: Final global random forest of Horizontal-PPGRF for Iris Dataset

IGRF-1	**IGRF-2**	**H-FGRF**
18	**1**	**19**
31	**4**	**35**
43	**4**	**47**
52	**7**	**59**
62	**12**	**74**
77	**10**	**87**
50	**16**	**66**
81	**17**	**98**
90	**15**	**105**
103	**18**	**121**

Table 5.3: IGRF-1 and IGRF-2 of Seeds Dataset in Horizontal-PPGRF

Trees	LRF-1	LRF-2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	9	4	**13**	1	1	**2**
20	12	13	**25**	4	3	**7**
30	30	19	**49**	0	3	**3**
40	32	22	**54**	4	7	**11**
50	47	27	**74**	1	9	**10**
60	58	30	**88**	2	11	**13**
70	61	42	**103**	4	9	**13**
80	73	46	**119**	4	12	**16**
90	77	40	**117**	8	17	**25**
100	80	52	**132**	11	14	**25**

Table 5.4: Final Global Random Forest of Seeds dataset

No of Trees	**IGRF-1**	**IGRF-2**	**H-FGRF**
10	13	2	**15**
20	25	7	**32**
30	49	3	**52**
40	54	11	**65**
50	74	10	**84**
60	88	13	**101**
70	103	13	**116**
80	119	16	**135**
90	117	25	**142**
100	132	25	**157**

Table 5.5: IGRF-1 and IGRF-2 of Clinical Dataset in Horizontal-PPGRF

Trees	LRF1	LRF2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	8	10	**18**	1	0	**1**
20	11	20	**31**	4	0	**4**
30	13	30	**43**	4	0	**4**
40	13	39	**52**	6	1	**7**
50	17	45	**62**	7	5	**12**
60	18	59	**77**	9	1	**10**
70	14	36	**50**	4	12	**16**
80	22	59	**81**	10	7	**17**
90	17	73	**90**	5	10	**15**
100	22	81	**103**	9	9	**18**

Table 5.6: Final Global Random Forest of Clinical Dataset in Horizontal-PPGRF

Trees	**IGRF-1**	**IGRF-2**	**H-FGRF**
10	18	1	19
20	31	4	35
30	43	4	47
40	52	7	59
50	62	12	74
60	77	10	87
70	50	16	66
80	81	17	98
90	90	15	105
100	103	18	121

Table 5.7: IGRF-1 and IGRF-2 of Glass dataset using Horizontal-PPGRF

Trees	LRF1	LRF2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	10	10	**20**	10	9	**19**
20	20	20	**40**	16	19	**35**
30	30	30	**60**	24	27	**51**
40	40	40	**80**	36	37	**73**
50	50	50	**100**	44	47	**91**
60	60	60	**120**	58	59	**117**
70	70	70	**140**	55	63	**118**
80	80	80	**160**	73	72	**145**
90	90	90	**180**	78	88	**166**
100	100	100	**200**	91	83	**174**

Table 5.8: Final Global Random Forest of Glass Dataset in Horizontal-PPGRF

Trees	**IGRF-1**	**IGRF-2**	**H-FGRF**
10	20	19	**19**
20	40	35	**35**
30	60	51	**51**
40	80	73	**73**
50	100	91	**91**
60	120	117	**117**
70	140	118	**118**
80	160	145	**145**
90	180	166	**166**
100	200	174	**174**

5.4.2 Time Complexity Analysis of Horizontal-PPGRF

Run times of each dataset when performing Random Forest Classification (Non Privacy Preserving) followed by Run times of Horizontal-PPGRF classification are presented in following table.

Table 5.9: Run times of Random Forest Classification (Non PP)

No of Trees	Iris	Seeds	Glass	Clinical
10	0.51	0.6	0.5	0.4
20	0.77	0.8	0.9	0.7
30	1.1	1.1	1.4	1.1
40	1.52	1.5	1.7	1.5
50	1.81	1.9	2.1	1.6
60	1.86	2.2	2.8	2.1
70	2.17	2.5	3.06	2.3
80	2.53	3	3.5	2.6
90	2.73	3.2	3.8	3.04
100	3.02	3.8	4.4	3.96

Table 5.10: Run times of Horizontal-PPGRF Classification

No of Trees	Iris	Seeds	Glass	Clinical
10	0.38	0.35	0.4	1.2
20	0.57	0.57	0.7	1.9
30	0.82	0.75	0.8	2.3
40	1.01	1.08	1.2	2.8
50	1.18	1.38	1.4	3.4
60	2.25	1.41	1.5	4.2
70	1.69	1.58	1.9	4.9
80	2.03	1.8	1.9	5.1
90	2.1	2.32	2.4	6.3
100	2.5	2.72	2.4	6.4

Figure 5.6: Runtime of Iris Dataset in Horizontal-PPGRF

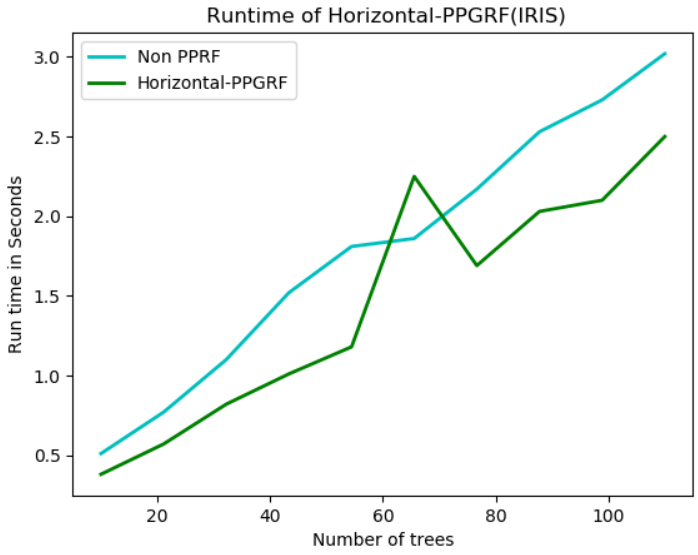


Figure 5.7: Runtime of Horizontal-PPGRF for Seeds Dataset

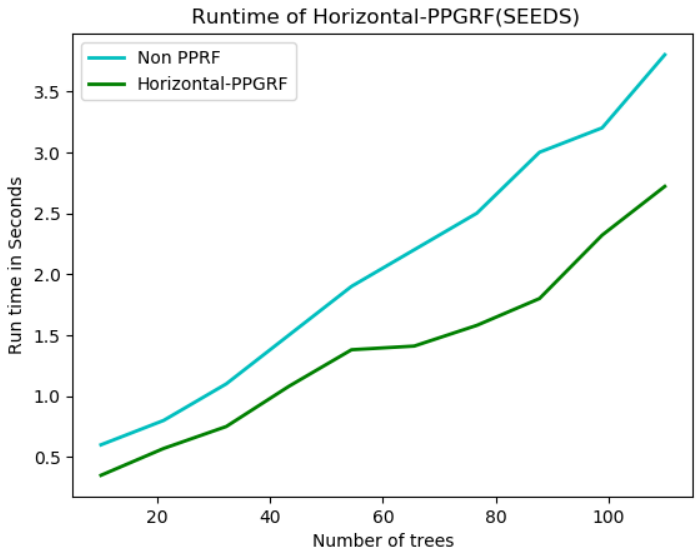


Figure 5.8: Runtime of Horizontal-PPGRF for Glass Dataset

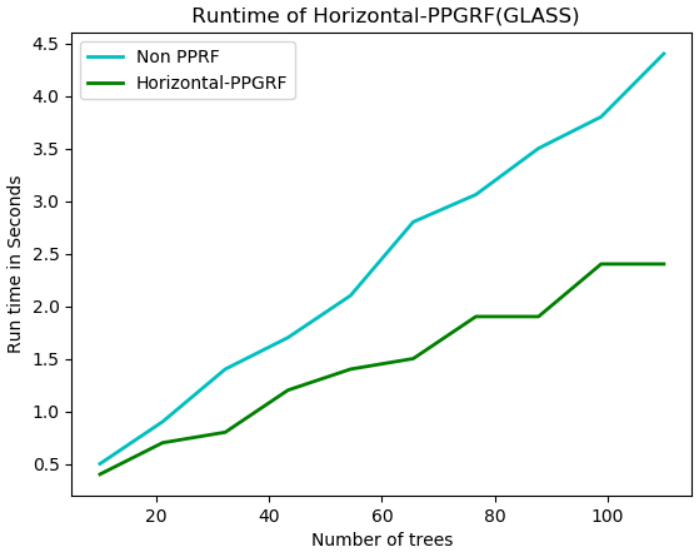
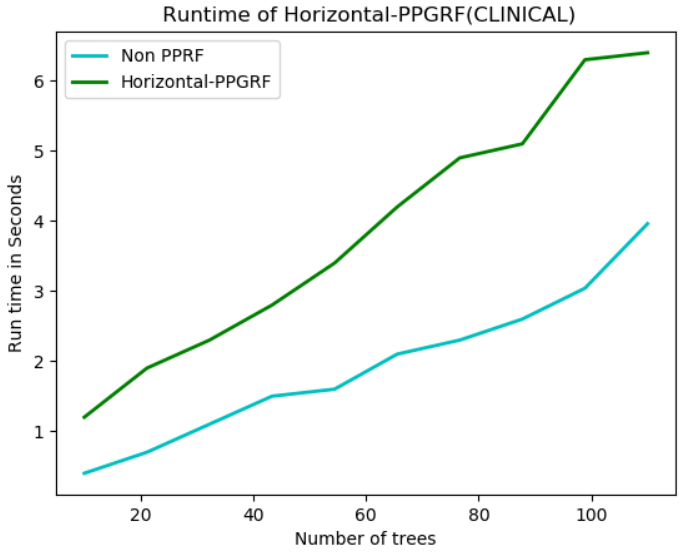


Figure 5.9: Runtime of Horizontal-PPGRF for Clinical dataset



5.4.3 Classification Accuracy of Horizontal-PPGRF

The accuracy scores are compared with non privacy preserving approach of building random forest. Horizontal-PPGRF accuracy scores are drawn for

Iris, Glass, Seeds and Clinical datasets are presented in following tables.

Table 5.11: Accuracy Scores of Non Privacy Preserving Random Forest

No of Trees	**Iris**	**Seeds**	**Glass**	**Clinical**
10	95	92	69	89
20	96	92	69	98
30	95	91	70	97
40	95	91	72	99
50	95	90	71	98
60	95	90	72	94
70	95	91	72	98
80	95	90	71	98
90	95	90	72	93
100	94	91	72	95

Table 5.12: Accuracy Scores of Party-1 in Horizontal-PPGRF

No of Trees	**Iris**	**Seeds**	**Glass**	**Clinical**
10	91	95	69	90
20	94	96	58	95
30	94	94	63	95
40	92	92	61	94
50	92	91	63	92
60	92	96	63	96
70	92	95	64	90
80	92	92	62	92
90	92	91	61	96
100	94	90	64	95

Table 5.13: Accuracy Scores of Party-2 in Horizontal-PPGRF

No of Trees	Iris	Seeds	Glass	Clinical
10	94	98	69	96
20	92	99	68	95
30	95	99	68	92
40	91	95	71	92
50	95	96	71	94
60	95	94	71	93
70	94	95	70	96
80	92	94	70	95
90	92	96	71	94
100	92	98	71	96

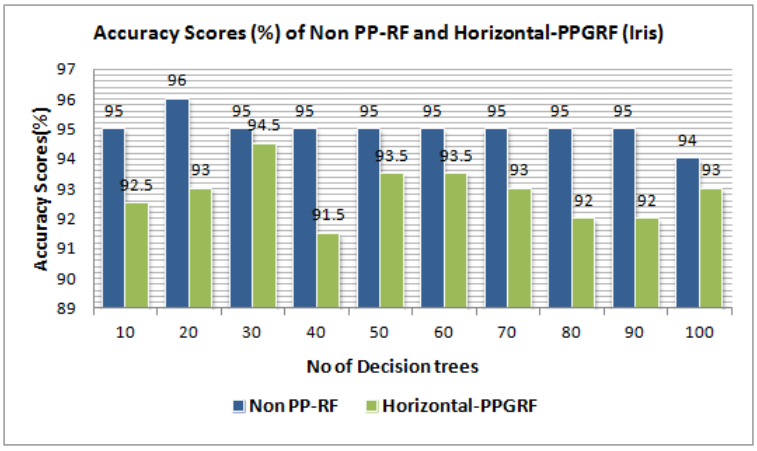


Figure 5.10: Accuracy of Horizontal-PPGRF Clustering for Iris Dataset

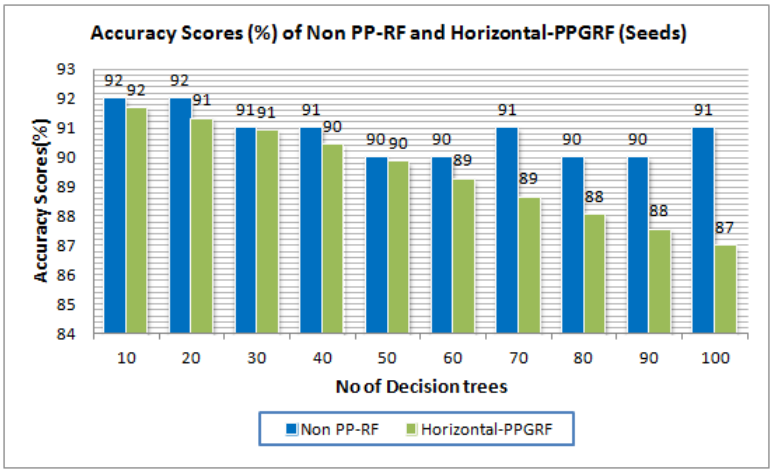


Figure 5.11: Accuracy of Horizontal-PPGRF Clustering for Glass Dataset

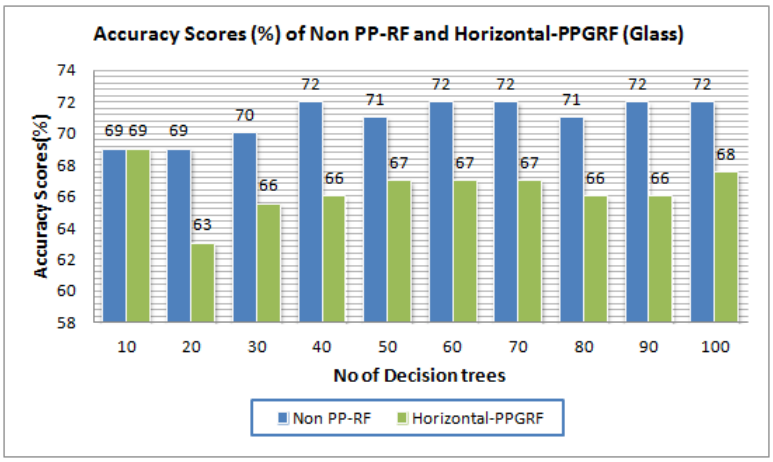


Figure 5.12: Accuracy of Horizontal-PPGRF Clustering for Seeds Dataset

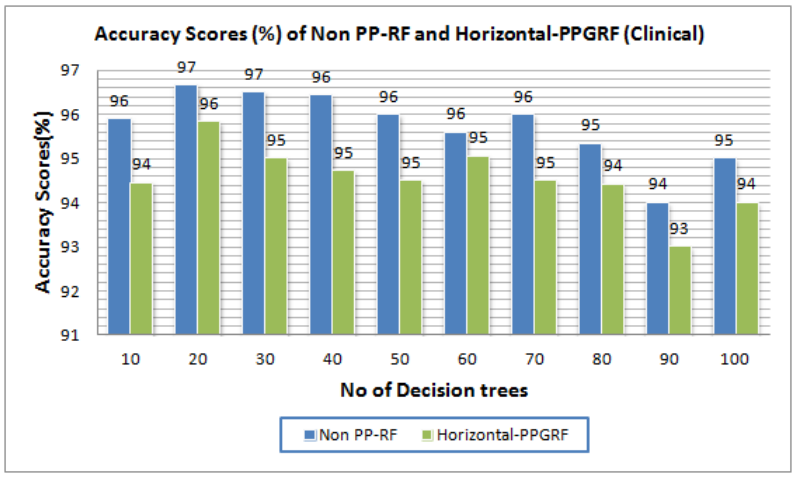


Figure 5.13: Accuracy of Horizontal-PPGRF Clustering for Clinical Dataset

5.4.4 Privacy Analysis of Horizontal-PPGRF

The privacy in Horizontal-PPGRF is evaluated using the privacy metrics given in chapter-2 and an analysis is done based on the results of privacy metrics. Privacy metrics and results for Horizontal-PPGRF are given in following table The privacy analysis is explained in view of three main phases

Table 5.14: Privacy Metrics Results and Analysis of Horizontal-PPGRF

Metric	**Range**	**Level**	Metric Result	Privacy Level
$priv_{CUE}$	$[0, \infty]$	$\geq$1.00=High	0.07	**Low**
$priv_{CMI}$	$[0, \infty]$	$\leq \infty$=Low	1.00	**High**
$priv_{PIC}$	[0, 1]	$\geq$0.5=High	0.05	**Low**

involved in the Horizontal-PPGRF method as follows

- ***Local level privacy:*** Local privacy is the major priority as parties have sensitive information at local level while building a global random forest. In proposed method no party directly share the raw data or the original attribute values to the other party, other than the resulting local random forest. Hence there will not be any privacy loss of an independent party. Result of privacy metric $priv_{PIC}$ (percentage incorrectly classified) derives the level of privacy at local level. In this case if incorrectly classified percentage of a party is high than the privacy level is high. The percentage is high for Horizontal-PPGRF and proved that the privacy is high.

- ***Aggregation level privacy:*** The privacy level must be high at this phase because, the aggregation is performed on the information shares by parties. Information or the results are produced by both the parties for the aggregation of LRF-1 and LRF-2 for building IGRF. Hence the level of privacy is the major concern and as per the results of privacy metric conditional mutual information $priv_{CMI}$ given the better privacy level (High) and shown that the aggregation level privacy is well preserved.

- ***Global level privacy:*** The final phase of building Global random forest must assure the global privacy, that means once parties shares their local level results and expects a secured outcome at global level. anyhow the voting is performed by local parties at their local level and will share only voted trees to the IGRF-1 and IGRF-2 for global aggregation. Hence the privacy is well preserved because of the secured aggregation without privacy violation. In this case also the privacy metric result has given the high privacy based on the cumulative entropy $priv_{CUE}$ value of both parties falls in high privacy range.

5.4.5 Results of Vertical-PPGRF

For each vertical partition, generates specified number of decision tress (10 to 100) limiting tree depth to 5. Constructs LRF-1 and LRF-2 considering all unique trees from the generated decision trees. Then the algorithm generates an Initial Global Random Forest (IGRF-1) by aggregation of all the unique trees from LRF-1 and LRF-2. Then all the unique decision trees with majority class from both parties to be selected through voting process, which results the two local random forests LRF-1A and LRF-2B from both parties. Then by aggregating and selecting unique trees from both parties, an Initial Global random Forest-2 (IGRF-2) will be constructed as shown in following table.

Now in last phase of the process a final global random forest is to be constructed by performing voting on initial global random forests (IGRF-1 & IGRF-2) generated by both the parties. The final aggregation of IGRF-1 and IGRF-2 takes place to build The final global random forest(FGRF-Vertical) which is the final global random forest constructed using privacy preserving random forest classification for vertical data distribution. In vertical case of iris dataset the party-2 returns the IGRF-2 as the final global random forest. The resulting table of Privacy Preserving Global Random Forest (FGRF-

Table 5.15: IGRF-1 and IGRF-2 of Vertical-PPGRF for Iris Dataset

Trees	LRF1	LRF2	**IGRF1**	Voting-1	Voting-2	**IGRF2**
10	10	10	**20**	9	8	**17**
20	20	20	**40**	8	15	**23**
30	30	30	**60**	17	19	**36**
40	40	40	**80**	20	22	**42**
50	50	50	**100**	23	34	**57**
60	60	60	**120**	26	28	**54**
70	70	70	**140**	33	55	**88**
80	80	80	**160**	43	23	**66**
90	90	90	**180**	44	42	**86**
100	100	100	**200**	56	88	**114**

Vertical) is given below.

Table 5.16: Final Global Random Forest in Vertical-PPGRF for Iris dataset

No.of Trees	**IGRF1**	**IGRF2**	**V-FGRF**
10	20	17	17
20	40	23	23
30	60	36	36
40	80	42	42
50	100	57	57
60	120	54	54
70	140	88	88
80	160	66	66
90	180	86	86
100	200	114	114

5.4.6 Time Complexity Analysis of Vertical-PPGRF

Run times of privacy preserving global random forest classification for vertical data distribution are given in below table.

Table 5.17: Results of Vertical-PPGRF for Seeds Dataset

Trees	LRF1	LRF2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	10	10	**20**	8	9	**17**
20	20	20	**40**	12	19	**31**
30	30	30	**60**	23	25	**48**
40	40	40	**80**	28	35	**63**
50	50	50	**100**	31	43	**74**
60	60	60	**120**	50	44	**94**
70	70	70	**140**	56	61	**117**
80	80	80	**160**	61	69	**130**
90	90	90	**180**	61	69	**130**
100	100	100	**200**	68	78	**146**

Table 5.18: Final Global random Forest of Vertical-PPGRF for Seeds Dataset

No of Trees	**IGRF-1**	**IGRF-2**	**V-FGRF**
10	20	17	17
20	40	31	31
30	60	48	48
40	80	63	63
50	100	74	74
60	120	94	94
70	140	117	117
80	160	130	130
90	180	130	130
100	200	146	146

Table 5.19: Results of Clinical dataset using Vertical-PPGRF Classification

Trees	LRF1	LRF2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	10	9	**19**	0	1	**1**
20	20	16	**36**	0	3	**3**
30	30	16	**46**	0	7	**7**
40	37	26	**63**	3	6	**9**
50	50	34	**84**	0	6	**6**
60	60	27	**87**	0	8	**8**
70	68	40	**108**	1	14	**15**
80	78	35	**113**	2	19	**21**
90	82	50	**132**	2	14	**16**
100	97	55	**152**	2	17	**19**

Table 5.20: Final Global random Forest of Clinical Dataset for Vertical-PPGRF

Trees	**IGRF-1**	**IGRF-2**	**V-FGRF**
10	19	1	20
20	36	3	39
30	46	7	53
40	63	9	72
50	84	6	90
60	87	8	95
70	108	15	123
80	113	21	134
90	132	16	148
100	152	19	171

Table 5.21: IGRF-1 and IGRF-2 of Glass dataset using Vertical-PPGRF

Trees	LRF1	LRF2	**IGRF-1**	Voting-1	Voting-2	**IGRF-2**
10	10	10	**20**	10	9	**19**
20	20	20	**40**	20	19	**39**
30	30	30	**60**	29	28	**57**
40	40	40	**80**	37	39	**76**
50	50	50	**100**	48	46	**94**
60	60	60	**120**	60	59	**119**
70	70	70	**140**	69	70	**139**
80	80	80	**160**	77	72	**149**
90	90	90	**180**	89	86	**175**
100	100	100	**200**	98	92	**190**

Table 5.22: Final Global Random Forest in Vertical-PPGRF for Glass Dataset

Trees	**IGRF-1**	**IGRF-2**	**V-FGRF**
10	20	19	**19**
20	40	39	**39**
30	60	57	**57**
40	80	76	**76**
50	100	94	**94**
60	120	119	**119**
70	140	139	**139**
80	160	149	**149**
90	180	175	**175**
100	200	190	**190**

Table 5.23: Run times of Vertical-PPGRF Classification

No of Trees	**Iris**	**Seeds**	**Glass**	**Clinical**
10	0.39	0.36	0.4	0.26
20	0.57	0.61	0.6	0.52
30	0.78	0.88	0.9	0.59
40	1.88	1.24	1.2	0.75
50	1.29	1.33	1.4	0.93
60	1.57	1.61	1.7	1.24
70	1.66	1.81	1.9	1.46
80	1.86	2.17	2.4	1.46
90	2.15	2.35	2.4	1.6
100	2.51	2.52	2.7	1.93

Figure 5.14: Runtime of Iris Dataset in Vertical-PPGRF

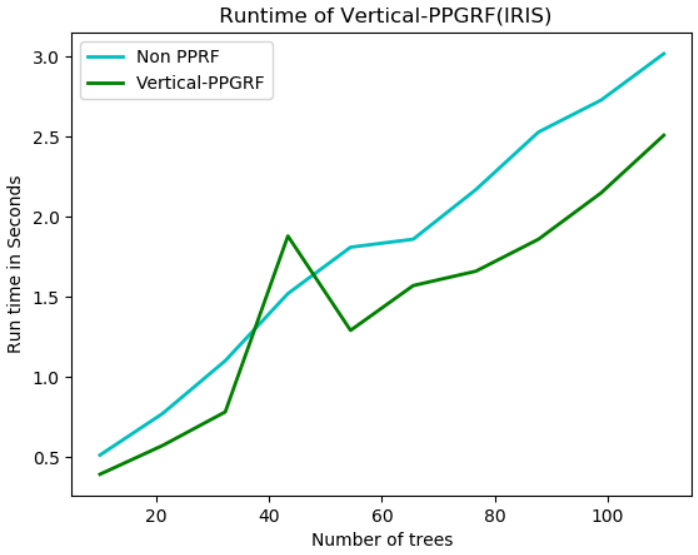


Figure 5.15: Runtime of Vertical-PPGRF for Seeds Dataset

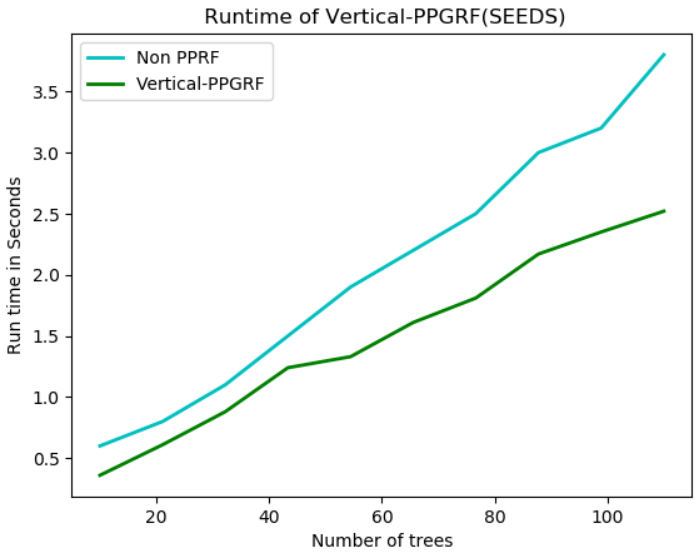


Figure 5.16: Runtime of Vertical-PPGRF for Glass Dataset

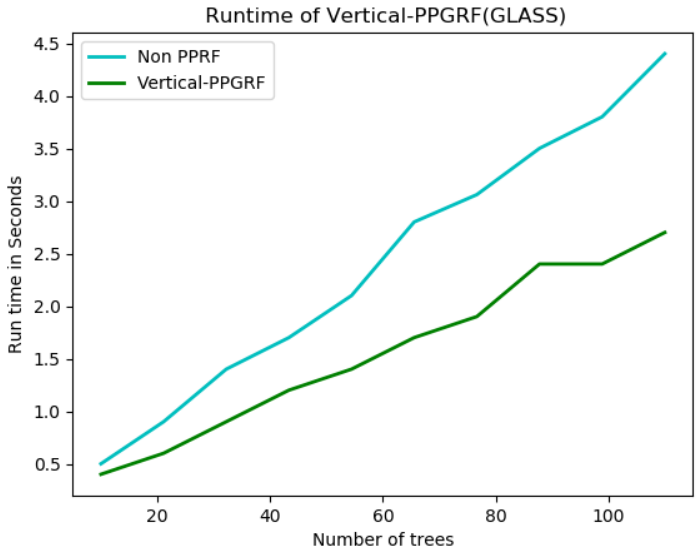
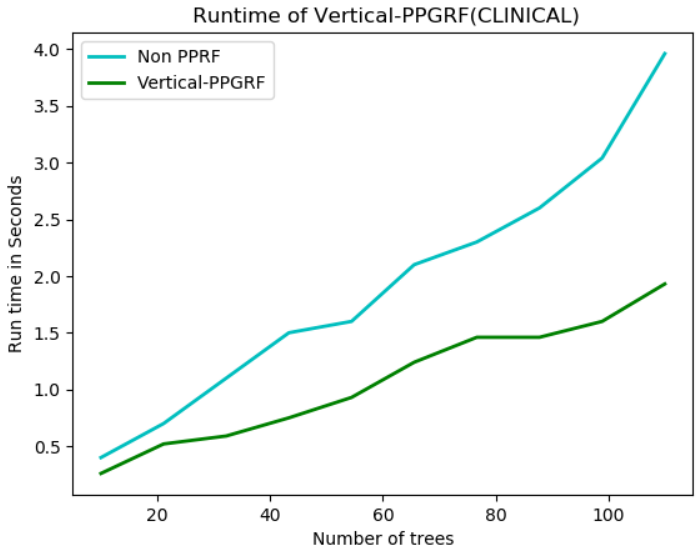


Figure 5.17: Runtime of Vertical-PPGRF for Clinical dataset



5.4.7 Classification Accuracy of Vertical-PPGRF

The accuracy percentages of initial Global random forests constructed by parties are presented in following tables and in respective graphs.

Table 5.24: Accuracy Scores of Non Privacy Preserving Random Forest

No of Trees	Iris	Seeds	Glass	Clinical
10	95	92	69	89
20	96	92	69	98
30	95	91	70	97
40	95	91	72	99
50	95	90	71	98
60	95	90	72	94
70	95	91	72	98
80	95	90	71	98
90	95	90	72	93
100	94	91	72	95

Table 5.25: Accuracy Scores of Party-1 in Vertical-PPGRF

No of Trees	Iris	Seeds	Glass	Clinical
10	76	85	69	85
20	76	85	72	85
30	75	85	72	85
40	74	86	73	87
50	75	86	72	87
60	75	85	72	87
70	75	85	72	87
80	75	85	72	87
90	75	84	72	87
100	74	84	73	87

Table 5.26: Accuracy Scores of Party-2 in Vertical-PPGRF

No of Trees	Iris	Seeds	Glass	Clinical
10	95	93	61	84
20	96	93	62	84
30	97	93	62	84
40	97	93	63	84
50	97	93	64	83
60	97	92	63	83
70	97	93	64	84
80	97	93	64	84
90	96	93	64	84
100	97	93	63	83

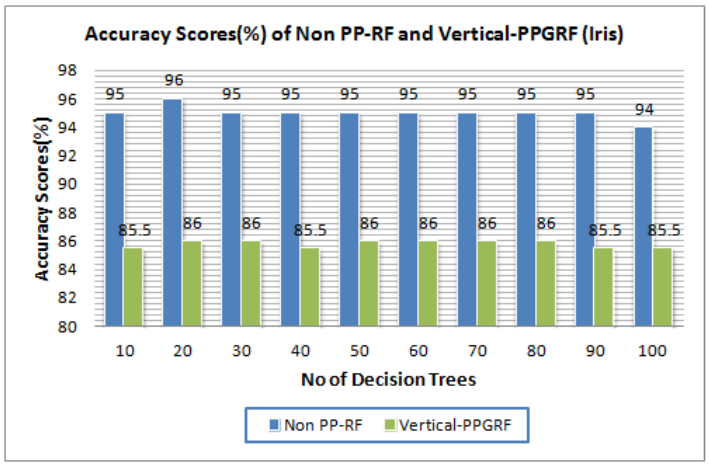


Figure 5.18: Vertical-PPGRF Accuracy scores for Iris Dataset

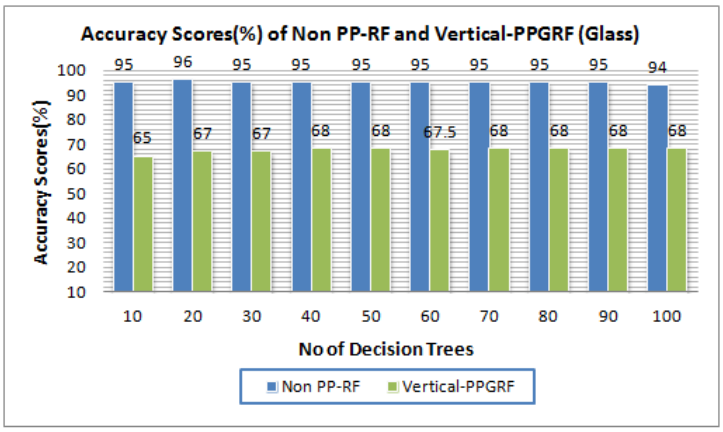


Figure 5.19: Vertical-PPGRF Accuracy scores for Glass Dataset

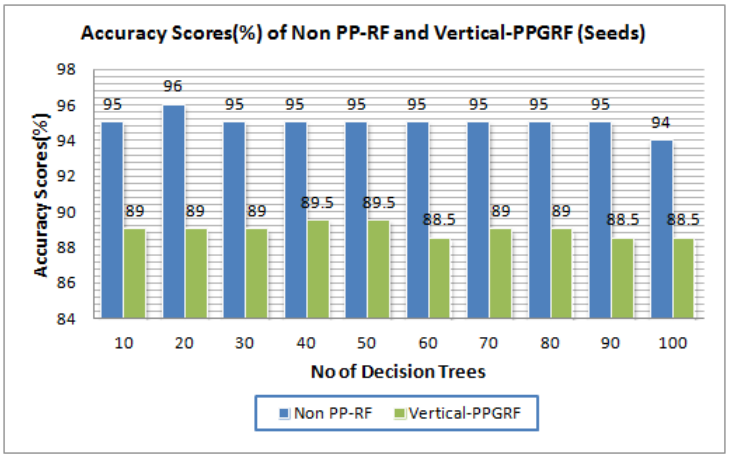


Figure 5.20: Vertical-PPGRF Accuracy scores for Seeds Dataset

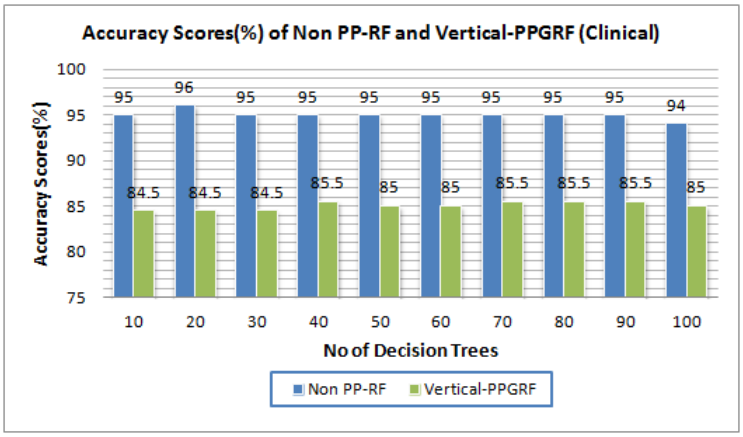


Figure 5.21: Vertical-PPGRF Accuracy scores for Clinical Dataset

5.4.8 Privacy Analysis of Vertical-PPGRF

The privacy analysis is presented for Vertical-PPGRF in this section. The results of privacy metrics for VPPGRF are presented in following table that includes the privacy level measured. Privacy analysis is explained with the help of privacy measures in view of three main phases of Vertical-PPGRF method as given below

- ***Local level privacy:*** In vertical partitioning case, the Local level privacy of independent parties is more important as parties have sensitive information at their level while building a global random forest. The

Table 5.27: Privacy Metrics Results of Vertical-PPGRF

Metric	**Range**	**Level**	Metric Result	Privacy Level
$priv_{CUE}$	[0, ∞]	$\geq$1.00=High	0.95	**Low**
$priv_{CMI}$	[0, ∞]	$\leq$1.00=Low	0.74	**High**
$priv_{PIC}$	[0, 1]	$\geq$0.5=High	0.33	**High**

proposed method gives independent path to each party to share the raw data or original values to the other party. Hence there will not be any privacy loss of information of an independent party. Result of privacy metric $priv_{PIC}$ (percentage incorrectly classified) derives the privacy of a party at local level with respect to the individual classification percentage. If incorrectly classified percentage of a party is high than the privacy level is said to be high. The Vertical-PPGRF method shows the high privacy level, hence the privacy is considered as well maintained.

- ***Aggregation level privacy:*** Privacy level in this intermediate phase must be high because of the aggregation of the information shared by parties. The results are produced by both the parties for the aggregation of LRF-1 and LRF-2 for building IGRF-1 and IGRF-2, hence a high level privacy is to be assured. As per the results of privacy metric used for this case, conditional mutual information $priv_{CMI}$ has given the better privacy level (High) and proved that the privacy is well preserved.

- ***Global level privacy:*** In process of building a final Global random forest, the global level privacy of parties must be assured by the model, because parties expect a secured outcome at global level after sharing local information. Though the voting is performed by local parties at their local level, they share voted trees to build the IGRF-1 and IGRF-2, hence the privacy is to be well maintained. The metric cumulative entropy $priv_{CUE}$ is used for this phase, that has given the high privacy.

5.4.9 Complexity analysis & Scalability of PPGRF:

In both Horizontal and Vertical PPGRF, datasets used are with instances up to 300 and computational complexity has shown high when number of

parties increased and number of secure computations are done in every level of communication. In Horizontal-PPGRF we use sequential collaboration for privacy preserving at the time of constructing local random forests and initial global random forests, hence the computational complexity increases for every level. In Vertical PPGRF we use the same way of collaboration approach and appending after voting is done, hence the computational overhead is increased and privacy is still preserved. When PPGRF methods tested for a larger dataset (cloud data with 2053 instances and 10 attributes), it could not score acceptable accuracy and privacy level.

5.5 Chapter Summary

The chapter titled Privacy Preserving Random Forest Classification, presented the method of building a global random forest by aggregating two local random forests of two local parties by a secured voting procedure. The chapter presented proposed algorithms for horizontal and vertical data distributions between parties to for random forests. The ensemble aggregation is the privacy preserving phase where the voting happens at each party separately, then the aggregation of decision trees will be done based on the majority of voting on the classes. The results were presented for both horizontal and vertical versions of Privacy Preserving Global Random Forest classification.

Chapter 6

Conlusions & Future Scope

6.1 Summary

The entire work has been undertaken for Horizontal and Vertical distribution of data set between multiple parties. Three major problems were presented, PPSOM, PPFCM and PPGRF. For first problem we adopted data perturbation technique for horizontal PPSOM and Cryptography based techniques for vertical PPSOM to perform clustering of two or more parties. The proposed algorithms for PPSOM are novel contributions of this chapter. Our experimental outcomes for both horizontal PPSOM and vertical PPSOM algorithms shows that the privacy is preserved when clustering is performed by multiple parties in distributed data environment.

For the first problem the work inspiration for horizontal PPSOM was taken from [20]. The initial work done on privacy preserving SOM clustering based on collaborative clustering has published in [14]. For vertical data distribution we adopted cryptography based methods to preserve privacy using SOM, inspired from the paper titled "Privacy preserving back-propagation neural network learning" [11] proposed by T Cheng and S Zhong. They presented Privacy preserving two party distributed back-propagation training algorithm to securely computing combined outputs in a neural network. We adopted the cryptography based approach used in their work for SOM to form clusters in vertically distributed data environment.

For second problem (PPFCM) adopted the core background work done by[32][33], and implemented a sequential collaboration when exchanging internal outputs between parties to perform collaborative clustering. Proposed algorithms for horizontal and vertical data distribution centric clustering. The modifications are mentioned in algorithms and results were produced for four different datasets. The aim of achieving privacy preserving, was

successful in collaborative clustering using Fuzzy C-Means Clustering.

For third problem the referral work taken from [4] and modified the stages as per the requirements to build a privacy preserving global random forest classifier. The process has two phases from local random forest to global random forest. The aim was to build a final global random forest by aggregating based on voting between parties. The proposed algorithm for horizontal and vertical were performing well in building PPGRF without disclosing the input information of any party. The bagging ensemble learning is applied through random forest classifier to build a secured global random forest.

6.2 Limitations

The thesis work presented privacy preserving methods including results and privacy concerns along with privacy metrics to measure the privacy of proposed algorithms. All the proposed algorithms were showing acceptable performance, accuracy and privacy levels. Some limitations are noticed based on various aspects like distribution of data, number of parties involved at the time of combined computations.

6.2.1 Limitations of PPSOM

Horizontal-PPSOM: used perturbation mechanism to preserve the attribute level privacy, but there is chance of unwanted distortion of values that leads to the loss of quality or originality of attribute values. The perturbation methods may not give accurate results compared to the other privacy preserving methods.
Vertical-PPSOM: used cryptography based approach to preserve privacy while exchanging information between parties. In this method though the privacy is very well preserved, there is high communication overhead because of number of encrypted & decrypted messages sent and received by the parties.

6.2.2 Limitations of PPFCM

Horizontal-PPFCM: uses sequential/parallel collaboration method, where impact of the collaboration is expressed in the changes of resulting centroids. Hence there is chance of overlap or redundancy of cluster center values, that leads to the multiple assignments or outliers of membership degrees.
Vertical-PPFCM: uses central collaboration method, where effect of collaboration is noticed in cluster centers also partition matrices of vertical par-

titions. In vertical scenario there is a chance of miss placing of members into clusters if the partial information is not correctly combined in collaboration.

6.2.3 Limitations of PPGRF

Horizontal-PPGRF: uses local and global ensemble aggregation & voting mechanism for building global random forest. The process may have high computational complexity and an increased miss classification percentage depending on the nature of dataset. Hence the model is to be generalized for larger datasets, where the global random forest may have high number of decision trees.
Vertical-PPGRF: uses the local and global level aggregation & voting mechanism for aggregating individual attribute values into the local random forests. Hence there is chance of redundancy in decision trees at local level and aggregation of redundant trees may lead to the entry of duplicate trees in global random forest. The model is to be generalized.

6.3 Future Scope

Our further investigations are aimed for increased number of parties for horizontal and vertical data distribution and also aimed to use arbitrary method of data partitioning. In view of growing privacy issues due to large amount of data distributed among multiple locations, there is necessity of building full secure models that are designed with assured privacy. Hence we are aiming to build a secure learning model where data exchange is possible without loosing privacy of any data site. There is always a great need of protecting data in various computing environments and there is always scope of developing and enhancing privacy preserving methods. There is scope of building time series privacy preserving methods helpful in medical, financial and defense applications. There is a larger scope in elaborating collaborative modes of communication and use combination of collaboration methods to increase the privacy and accuracy level. Finally there is a large scope of developing privacy metrics based on process and results of privacy preserving methods.

List of Publications from Thesis Work

- Gadepaka, Latha, and Bapi Raju Surampudi. "Privacy Preserving Collaborative Clustering Using SOM for Horizontal Data Distribution." In Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015), pp. 273-284. Springer, Cham, 2015.

- Gadepaka Latha, Surampudi Bapi Raju. "Privacy Preserving and Secured Clustering of Distributed Data using Self Organizing Map". 1st International Conference on Cyber Warfare, Security & Space Research. SpacSec-2021, Manipal University, Jaipur, India, 9-10 December 2021.

- Latha Gadepaka, and Bapi Raju Surampudi. "Privacy Preserving of Two Local Data Sites using Global Random Forest Classification". 1st International Conference on Cyber Warfare, Security & Space Research. SpacSec-2021, Manipal University, Jaipur, India, 9-10 December 2021.

- Latha Gadepaka, and Bapi Raju Surampudi. "Privacy preserving of two collaborating parties using Fuzzy C-Means Clustering". International Conference on Advanced Network Technologies and Intelligent Computing. ANTIC-2021, Banaras Hindu University, Varanasi, India, 17-18 December 2021.

- Latha Gadepaka, and Bapi Raju Surampudi. Evaluation of Privacy level using Privacy Evaluation Metrics in Privacy Preserving classification and Clustering Methods."International Journal of Computational Intelligence Systems, (ISSN-1875-6883).

Bibliography

[1] D. Agrawal and R. Srikant. Privacy preserving data mining. *In Proc. ACM SIGMOD*, pages 439–450, 2000.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, May 2000.

[3] Alia Alabdulkarim, Mznah Al-Rodhaan, Yuan Tian, and Abdullah Al-Dhelaan. A privacy-preserving algorithm for clinical decision-support systems using random forest. *CMC Comput. Mater. Con*, 58:585–601, 2019.

[4] Yuan Tian Abdullah Al-Dhelaan Alia Alabdulkarim, Mznah Al-Rodhaan. A privacy-preserving algorithm for clinical decision-support systems using random forest. *Computers, Materials & Continua*, 58(3):585–601, 2019.

[5] Kellie J Archer and Ryan V Kimes. Empirical characterization of random forest variable importance measures. *Computational statistics & data analysis*, 52(4):2249–2260, 2008.

[6] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

[7] Alper Bilge and Huseyin Polat. A comparison of clustering-based privacy-preserving collaborative filtering schemes. *Applied Soft Computing*, 13(5):2478–2489, 2013.

[8] Alper Bilge and Huseyin Polat. A comparison of clustering-based privacy-preserving collaborative filtering schemes. *Applied Soft Computing*, 13(5):2478–2489, 2013.

[9] C. L. Blake and C. J. Merz. Uci repository of machine learning databases. *University of California, Department of Information and Computer Science*, 1998.

[10] Matthieu Bloch, Onur Günlü, Aylin Yener, Frédérique Oggier, H Vincent Poor, Lalitha Sankar, and Rafael F Schaefer. An overview of information-theoretic security and privacy: Metrics, limits and applications. *IEEE Journal on Selected Areas in Information Theory*, 2(1):5–22, 2021.

[11] T. Chen and S. Zhong. Privacy-preserving backpropagation neural network learning. *IEEE Transactions on Neural Networks*, 20(10):1554–1564, Oct 2009.

[12] Vladimir Estivill-Castro. Private representative-based clustering for vertically partitioned data. In *Computer Science, 2004. ENC 2004. Proceedings of the Fifth Mexican International Conference in*, pages 160–167. IEEE, 2004.

[13] M. Feingold, M. Corzine, M. Wyden, and M. Nelson. Data mining moratorium act of 2003. *U.S Senate Bill (proposed)*, 2003.

[14] Latha Gadepaka and Bapi Surampudi. *Privacy Preserving Collaborative Clustering Using SOM for Horizontal Data Distribution*, pages 273–284. 11 2015.

[15] F Gorgonio and J Costa. Privacy-preserving clustering on distributed databases: A review and some contributions. *Self Organizing Maps-Applications and Novel Algorithm Design, InTech*, pages 33–54, 2011.

[16] Flavius L Gorgônio and Jose Alfredo F Costa. Parallel self-organizing maps with application in clustering distributed data. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 3276–3283. IEEE, 2008.

[17] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[18] HIPPA. National standards to protect the privacy of personal health information. *http://www.hhs.gov/ocr/hipaa/finalreg.html*.

[19] C. Kaleli and H. Polat. Som-based recommendations with privacy on multi-party vertically distributed data. *Journal of the Operational Research Society*, 63(6):826–838, Jun 2012.

[20] Cihan Kaleli and Huseyin Polat. Privacy-preserving som-based recommendations on horizontally distributed data. *Knowledge-Based Systems*, 33:124–135, 2012.

[21] M. Kantarcioglu and C. Clifton. Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. In *ACM SIGMOD Workshop on Research Issues on DMKD'02*, June 2002.

[22] Fatemeh Khodaparast, Mina Sheikhalishahi, Hassan Haghighi, and Fabio Martinelli. Privacy preserving random decision tree classification over horizontally and vertically partitioned data. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 600–607, Aug 2018.

[23] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384, 1996.

[24] Pradeep Kumar, Kishore Indukuri Varma, and Ashish Sureka. Fuzzy based clustering algorithm for privacy preserving data mining. *International Journal of Business Information Systems*, 7(1):27–40, 2011.

[25] Ming Li, Lanlan Wang, and Haiju Fan. Privacy-preserved data hiding using compressive sensing and fuzzy c-means clustering. *International Journal of Distributed Sensor Networks*, 16(2):1550147720908748, 2020.

[26] Qiongxiu Li, Jaron Skovsted Gundersen, Richard Heusdens, and Mads Græsbøll Christensen. Privacy-preserving distributed processing: Metrics, bounds and algorithms. *IEEE Transactions on Information Forensics and Security*, 16:2090–2103, 2021.

[27] X. Lin, C. Clifton, and M. Zhu. Privacy Preserving Clustering with Distributed EM Mixture Modeling. *Knowledge and Information Systems*, to appear.

[28] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.

[29] V Manikandan, V Porkodi, Amin Mohammed, and M.Sivaram Murugan. Privacy preserving data mining using threshold based fuzzy c-means clustering. 9:1820–1823, 10 2018.

[30] Ricardo Mendes and João P Vilela. Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017.

[31] Yuichi Nakamura, Keiya Harada, and Hiroaki Nishi. A privacy-preserving sharing method of electricity usage using self-organizing map. *ICT Express*, 4(1):24–29, 2018.

[32] Witold Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23(14):1675–1686, 2002.

[33] WITOLD PEDRYCZ. Distributed and collaborative fuzzy modeling. 2007.

[34] Witold Pedrycz and Partab Rai. Collaborative clustering with the use of fuzzy c-means and its quantification. *Fuzzy Sets and Systems*, 159(18):2399–2427, 2008.

[35] Witold Pedrycz and George Vukovich. Clustering in the framework of collaborative agents. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, volume 1, pages 134–138. IEEE, 2002.

[36] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[37] Gábor Szucs. Decision trees and random forest for privacy-preserving data mining. In *Research and Development in E-Business through Service-Oriented Solutions*, pages 71–90. IGI Global, 2013.

[38] Jaideep Vaidya, Christopher W Clifton, and Yu Michael Zhu. *Privacy preserving data mining*, volume 19. Springer Science & Business Media, 2006.

[39] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):1–38, 2018.

[40] A. C. Yao. Protocols for secure computations. *In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[41] Fusheng Yu, Juan Tang, and Ruiqiong Cai. Partially horizontal collaborative fuzzy c-means. *International Journal of Fuzzy Systems*, 9(4):198, 2007.

[42] Sheng Zhang, James Ford, and Fillia Makedon. A privacy-preserving collaborative filtering scheme with two-way communication. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 316–323. ACM, 2006.

[43] Jie Zhou, Witold Pedrycz, Xiaodong Yue, Can Gao, Zhihui Lai, and Jun Wan. Projected fuzzy c-means clustering with locality preservation. *Pattern Recognition*, 113:107748, 2021.

Appendix A

A.1 Privacy Preserving Collaborative Clustering using SOM

Algorithm 11 Privacy Preserving Horizontal Collaborative SOM Algorithm

1. Dataset is horizontally partitioned between n parties, then decide on c number of clusters, and determine the sequence of active parties to be followed in clustering process.
2. First Initial Party(say p_1) is active and it assigns w_j vectors for all c clusters and initialize required parameters.
3. Now IP(p_1) select a random user among all users it holds and perform clustering by deciding winner neuron, then updates weights.
4. p_1 repeats step 2 and 3 unless all it's users are allocated to a cluster, then sends the final w_j vectors and increased s value to next party(say p_2).
5. Now the present active party p_2 repeats step 2 and 3 same as p_1, until all it's users are allocated to a cluster, then it sends new s value and updated w_j vectors to the next party.
6. After all users are allocated to clusters, the last party updates w_j vectors and send to the IP(p_1).
7. Setps 3 to 6 are continued until no noticeable change in the SOM.

In privacy preserving collaborative SOM clustering (PPCSOM) method, first the dataset is horizontally partitioned and distributed between n number

of parties to form C number of clusters, with which each data partition is owned and resides at each party, then all parties cluster their own data while collaborating with each other without directly revealing their private information.

A.1.1 Results of Horizontal-PPSOM

Runtime(in seconds)	IRIS	Glass	Wine	Pima Indians
SOM Runtime	2.03	4.23	5.62	10.23
Horizontal-PPSOM Runtime	3.92	6.23	9.12	15.97

Table A.1: SOM Runtime and Horizontal-PPSOM Runtime

Figure A.1: Runtime of SOM Compared with Runtime of Horizontal-PPSOM

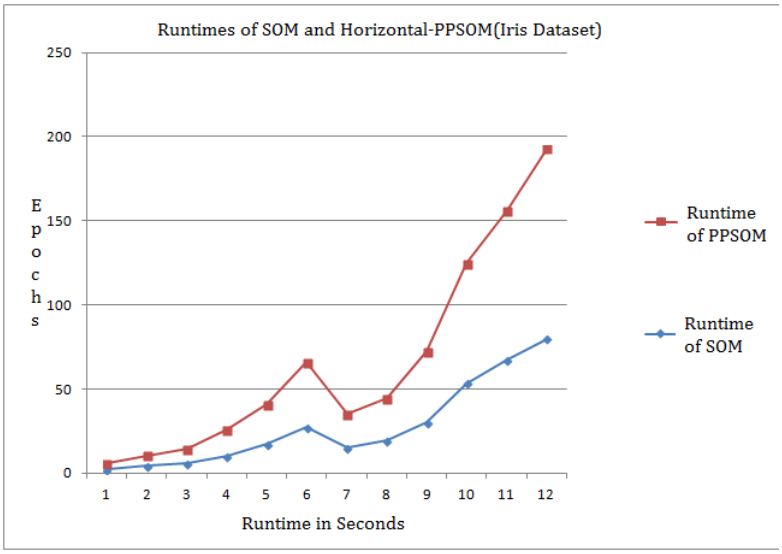


Figure A.2: Accuracy of SOM and PPCSOM

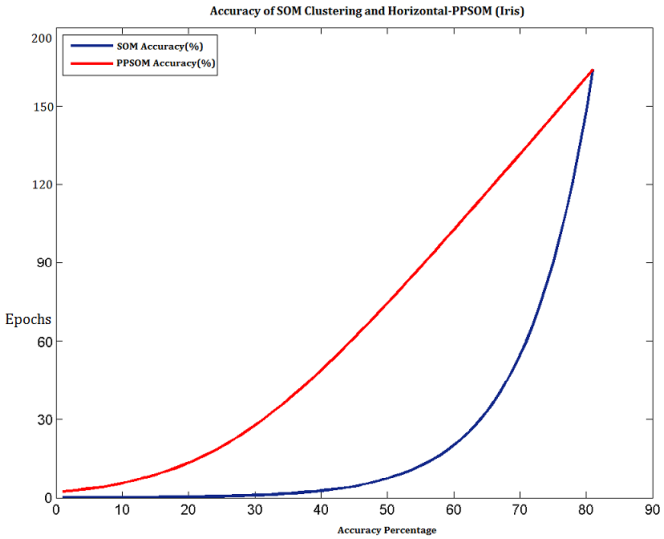


Figure A.3: A Poster on Privacy Preserving Collaborative Clustering in Horizontal-PPSOM

Privacy Preserving Collaborative Clustering using SOM for Horizontal Data Distribution

22-01-2016

Latha Gadepaka and Bapi Raju Surampudi

Abstract:

In view of present advancements in computing, with the development of distributed environment, many problems have to deal with distributed input data where individual data privacy is the most important issue to be addressed, for the concern of data owner by extending the privacy preserving notion to the original learning algorithms. Privacy Preserving Data Mining has become an active research area in addressing various privacy issues while bringing out solutions for them. There has been lot of progress in developing secure algorithms and models, able to preserve privacy using various data mining techniques like association, classification and clustering, where as importance of privacy preserving techniques applied for learning algorithms related to neural networks for mining problems are still in infancy. We focused on preserving privacy of an individual, using self organizing map (SOM) adopted for collaborative clustering of distributed data between multiple parties. We present Privacy Preserving Collaborative Clustering method using SOM (PPCSOM) for Horizontal Data Distribution, which allows multiple parties perform clustering in a collaborative approach using SOM neural network, without revealing their data directly to each other, in order to preserve privacy of all parties.

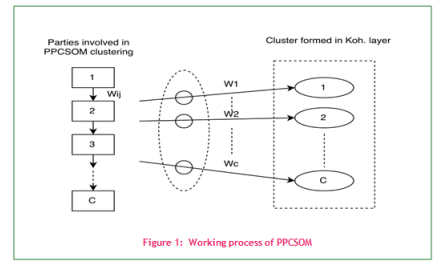


Figure 1: Working process of PPCSOM

Main Objectives:

- Privacy Preserving Collaborative Clustering method using SOM for Horizontal Data Distribution (PPCSOM)
- Allows multiple parties to perform clustering in a collaborative approach using SOM neural network
- Preserving privacy without revealing their data directly to each other
- Preserving privacy of an individual, of distributed data between multiple parties
- Performance evaluation of the method while Preserving privacy of all parties for horizontal data distribution

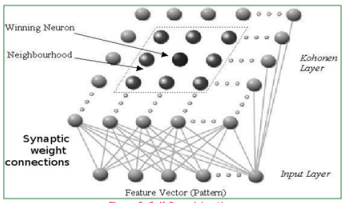


Figure 2: Self Organizing Map.

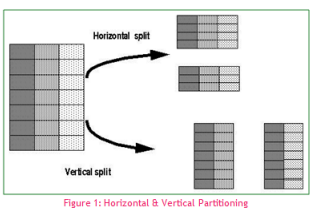


Figure 1: Horizontal & Vertical Partitioning

Main References:

- J. Vaidya, C. W. Clifton, and Y. M. Zhu. Privacy preserving data mining, volume 19. Springer Science & Business Media, 2006.
- C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, , and M. Zhu. Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations, 4(2), December 2002.
- C. Kaleli and H. Polat. Privacy-preserving som-based recommendations on horizontally distributed data. Knowledge-Based Systems, 33:124{135, 2012.
- S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.
- Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. Journal of Privacy and Condentiality, 1(1):5, 2009.

Working Process of PPCSOM:

- Privacy Preserving Collaborative Clustering method using SOM for Horizontal Data Distribution (PPCSOM)
- Allows multiple parties to perform clustering in a collaborative approach using SOM neural network
- Preserving privacy without revealing their data directly to each other
- Preserving privacy of an individual, of distributed data between multiple parties
- Performance evaluation of the method while Preserving privacy of all parties for horizontal data distribution

Results Analysis:

- PPCSOM helps multiple parties collaboratively construct clusters, while data is distributed in number of horizontal partitions.
- Better accuracy with an assured privacy and less communication overhead though at the cost of loss in accuracy.
- The same work can be applied for vertical and arbitrary ways of data partitioning.
- Collaborations in PPCSOM can be enhanced in future for better online performance.
- Vertical partitioning would require cryptographic approach to secretly share the information among number of parties.

Conclusions & Future Scope:

- PPCSOM helps multiple parties collaboratively construct clusters, while data is distributed in number of horizontal partitions.
- Better accuracy with an assured privacy and less communication overhead though at the cost of loss in accuracy.
- The same work can be applied for vertical and arbitrary ways of data partitioning.
- Collaborations in PPCSOM can be enhanced in future for better online performance.
- Vertical partitioning would require cryptographic approach to secretly share the information among number of parties.

A.2 Privacy Preserving Horizontal-ID3

ID3 algorithm is used to build decision Tree, which is a recursive process, which results a tree of decisions on attributes and class labels of dataset. A privacy preserving ID3 algorithm is used to to build a privacy preserved decision tree, with decision attributes "R" class attributes "C", and training entities "T". Perturbation based Privacy Preserving ID3 Algorithm is given below.

Algorithm 12 Privacy Preserving Horizontal-ID3 Algorithm

Partition: Data set is horizontally partitioned
Class Label: is known for all parties(two parties in this case)
Perturbing: Data is perturbed by adding random noise from a distribution
Recurse:
Step 1a: Find attributes A with highest information gain for all training samples T
Step 1b: Partition T based on values a_i of A
Step 1c: Return a decision Tree with root labeled A and edges a_i, with node at the end of edge a_i
Terminate: when all training samples are classified

A.2.1 Results of Horizontal-PPID3

Figure A.5: Decision Tree with perturbed Inputs of Iris Dataset

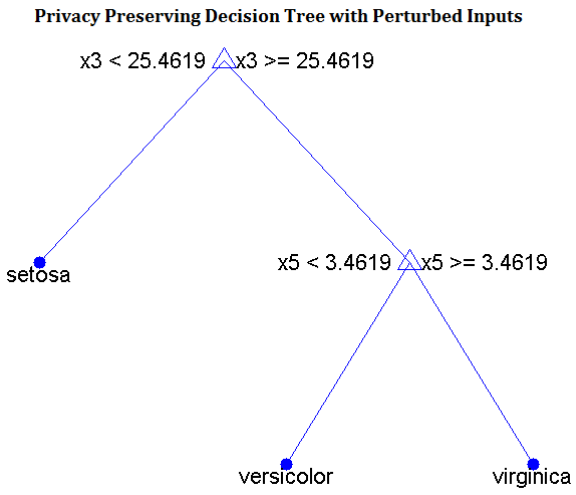


Figure A.4: Decision Tree without perturbed Inputs of Iris Dataset

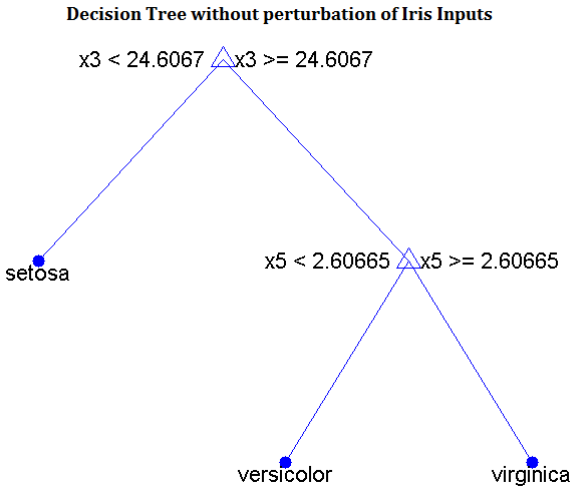


Figure A.7: Probability Distribution of Perturbed Inputs in Privacy Preserving Decision Tree

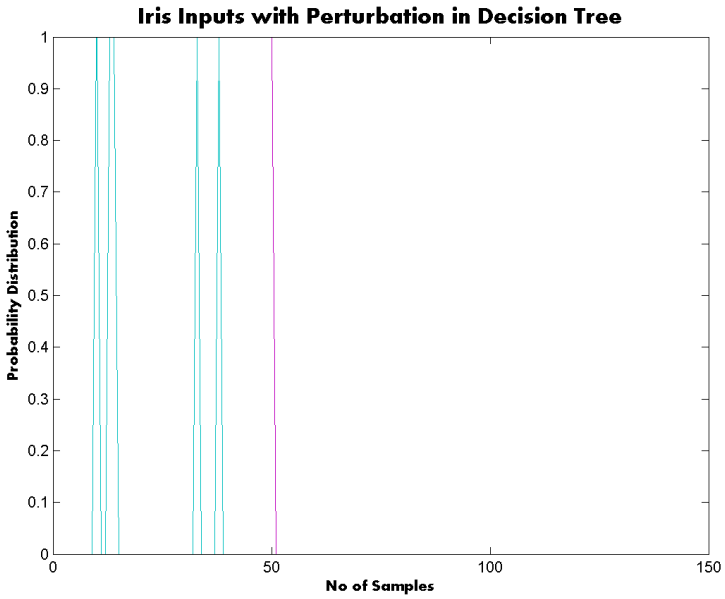
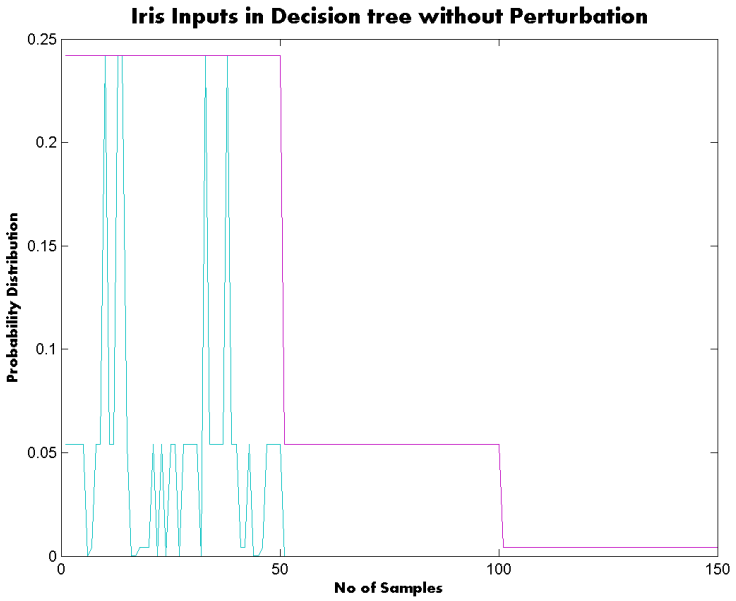


Figure A.6: Probability Distribution of Inputs in Decision Tree



A.3 Privacy Preserving Horizontal-Random Forest Classification

Privacy Preserving random forest classification is the novel approach to preserve privacy of distributed data while building the random forest of multiple decision trees. The privacy is preserved at input level(decision tree level) using ID3 algorithm on training dataset of two parties. The algorithm follows the process of adding noise to the input data and learning the model from noisy data set to build random forest of privacy preserved decision trees. The Horizontal-PPID3 Algorithm is given below The input attributes are

Algorithm 13 Privacy Preserving Horizontal-PPRF Algorithm

Partition: Data set is horizontally partitioned and class label is known for both parties
Perturbing: Data is perturbed by adding random noise from a normal distribution
Recurse: Call ID3 Decision Tree Algorithm (Horizontal-PPID3)
Local Random Forest: Party 1 & 2 build LRF-1 and LRF-2 at their sites.
Privacy Preserving Random Forest: Party-1 and Party-2 build PPRF
Terminate: If all samples are classified into Random forest

perturbed so that no one can know which entity belongs to which party (partition). Split points of noisy data are not obvious hence this problems can be solved by knowing the distribution of the original data, though the model dont know the original values.

A.3.1 Results of Horizontal-PPRF

Table A.2: Random Forest of Horizontal-PPRF for Iris Dataset

Trees	LRF-1	LRF-2	**Non PPRF**	PPLRF-1	PPLRF-2	**PPRF**
10	8	10	**18**	10	10	**20**
20	11	16	**27**	15	20	**35**
30	12	28	**40**	14	23	**37**
40	12	32	**44**	18	39	**57**
50	17	40	**57**	19	49	**68**
60	18	54	**72**	23	58	**81**
70	18	52	**70**	27	67	**94**
80	16	51	**67**	26	71	**97**
90	21	37	**58**	24	77	**101**
100	21	70	**91**	27	82	**109**

Table A.3: Runtime of Horizontal-PPRF for Iris Dataset

No of Trees	Non PP RF	PPRF
10	0.38	0.96
20	0.57	1.13
30	0.82	1.65
40	1.01	1.55
50	1.18	1.75
60	2.25	2.07
70	1.69	2.2
80	2.03	3.58
90	2.1	2.54
100	2.5	2.99

Table A.4: Accuracy Scores of Party-1 in Horizontal-PPRF for Iris Dataset

No of Trees	Non PPRF	PPRF
10	95	91
20	96	94
30	6	94
40	96	92
50	93	92
60	95	92
70	92	92
80	95	92
90	95	92
100	94	94

Table A.5: Accuracy Scores of Party-2 in Horizontal-PPRF for Iris Dataset

No of Trees	Non PP RF	PPRF
10	94	94
20	93	92
30	92	95
40	94	91
50	95	94
60	92	90
70	94	94
80	92	92
90	92	92
100	92	92

Figure A.8: Accuracy Scores of Party-1 and Party-2 in Horizontal-PPRF for Iris Dataset

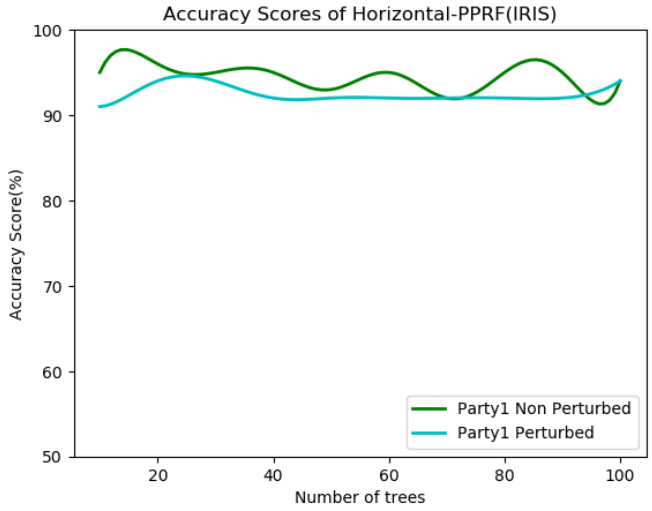


Figure A.9: Accuracy Scores of Party-1 and Party-2 in Horizontal-PPRF for Iris Dataset

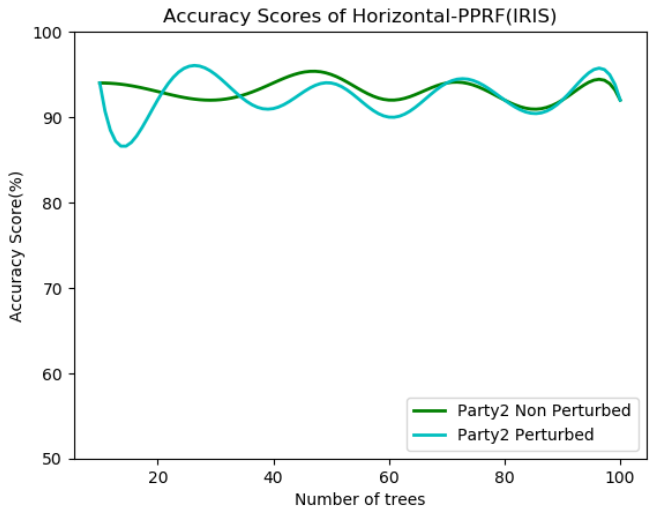
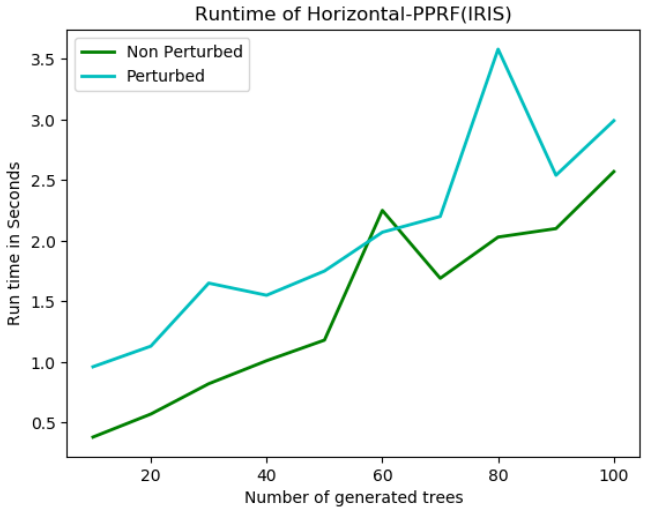


Figure A.10: Runtime Comparison graph for Non PPRF and Horizontal-PPRF



A.4 PhD Work (Thesis) - Summary Table

Complete PhD work - Summary Table							
SNo	**Privacy Preserving Technique/Method**	**Data Mining Functionality**	**Problem Domain**	**PP Approach**	**Number of Parties**	**Experiments Platform**	**Datasets Used**
1	Horizontal PPSOM	Clustering	Neural Networks	Perturbation	Two Party	Matlab +Weka	Iris, Glass, Wine, Seeds
2	Vertical PPSOM	Clustering	Neural Networks	Cryptography	Two Party	Matlab + C#	Iris, Glass, Wine, Seeds
3	Horizontal PPFCM	Clustering	Fuzzy Sets	Collaboration	Multi Party	Matlab + Fuzzy	Iris, Glass, Wine, Seeds
4	Vertical PPFCM	Clustering	Fuzzy Sets	Collaboration	Multi Party	Matlab + Fuzzy	Iris, Glass, Wine, Seeds
5	Horizontal PPRF	Classification	Ensemble Learning	Bagging	Two Party	Python + anaconda	Iris, Glass, Seeds, Clinical
6	Vertical PPRF	Classification	Ensemble Learning	Bagging	Two Party	Python + anaconda	Iris, Glass, Seeds, Clinical
7	*PPSOM	Clustering	Neural Networks	Collaboration	Multi Party	Matlab + SOM Tool box	Iris, Glass, Wine, Seeds
8	*PPID3	Classification	Machine Learning	Perturbation	Two Party	Matlab + SOM Tool box	Iris, Glass, Wine, Seeds
9	*PPRF	Classification	Ensemble Learning	Bagging	Multi Party	Python + Matlab	Iris, Glass, Seeds, Clinical
*** Additional work done**							

Abbreviations

PP	- Privacy Preserving
PPDM	- Privacy Preserving Data Mining
PPDDM	- Privacy Preserving Distributed Data Mining
SMC	- Secure Multi Party Computing
HP	- Horizontally partitioned
VP	- Vertically Partitioned
AP	- Arbitrarily Partitioned
GP	- Grid Partitioned
TP	- Two party
MP	- Multi Party
TTP	- Trust Third Party Model
SH	- Semi-honest Model
MM	- Malicious Model
OM	- Other Models
PPSOM	- Privacy Preserving Self Organizing Map
FCM	- Fuzzy C-Means Clustering
CC	- Collaborative Clustering
PPFCM	- Privacy Preserving Fuzzy C-Means Clustering
RF	- Random Forest
PPRF	- Privacy Preserving Random Forest
PPGRF	- Privacy Preserving Global Random Forest
LRF	- Local Random Forest
IGRF	- Initial Global random Forest
FGRF	- Final Global random Forest

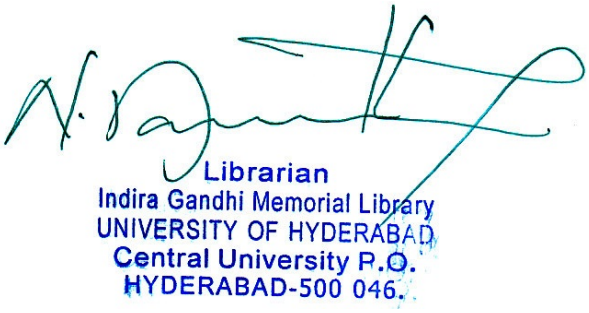
Table A.6: Privacy measuring notations used for PP Methods

d() = Distance function	D = Data Set
E = Equivalent Class	H() = Entropy
I(;) = Mutual Information	K = Privacy Mechanism
L = Location/Site	M = Messages/Requests
p(x) = p(X = x)	R = Regions
S = Sensitive attribute value	T = Time
U = Set of users u$\in$U	X = Discrete random variable
X^* = True distribution of hidden data	Y = Data observed by other party
Z = Prior Information	$\beta()$ = Loss Function
τ = Thresholds	ω = Weight

End of the Thesis

Investigation of Privacy Preserving Methods for Classification and Clustering of Distributed Data

by Latha Gadepaka



End of the Thesis

Investigation of Privacy Preserving Methods for Classification and Clustering of Distributed Data

ORIGINALITY REPORT

6%	3%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hyderabad, Hyderabad Student Paper	1%
2	Isabel Wagner, David Eckhoff. "Technical Privacy Metrics", ACM Computing Surveys, 2018 Publication	1%
3	"Privacy Preserving Collaborative Clustering Using SOM for Horizontal Data Distribution", Advances in Intelligent Systems and Computing, 2015. Publication	1%
4	cse.buffalo.edu Internet Source	<1%
5	link.springer.com Internet Source	<1%
6	pure.qub.ac.uk Internet Source	<1%
7	journals.usb.ac.ir Internet Source	

		<1%
8	epdf.tips Internet Source	<1%
9	Tobiszewski, Marek, Stefan Tsakovski, Vasil Simeonov, and Jacek Namieśnik. "Multivariate statistical comparison of analytical procedures for benzene and phenol determination with respect to their environmental impact", Talanta, 2014. Publication	<1%
10	Pedrycz, W.. "Collaborative clustering with the use of Fuzzy C-Means and its quantification", Fuzzy Sets and Systems, 20080916 Publication	<1%
11	Advances in Database Systems, 2008. Publication	<1%
12	mafiadoc.com Internet Source	<1%
13	Chin-Teng Lin, Mukesh Prasad, Jyh-Yeong Chang. "Designing mamdani type fuzzy rule using a collaborative FCM scheme", 2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY), 2013 Publication	<1%

14	"Privacy Preserving Data Mining", Springer Nature, 2006 Publication	<1 %
15	Sara Hajian, Mohammad Abdollahi Azgomi. "A privacy preserving clustering technique for horizontally and vertically distributed datasets", Intelligent Data Analysis, 2011 Publication	<1 %
16	www.tinbergen.nl Internet Source	<1 %
17	Planchon, F.A.. "Short-term variations in the occurrence of heavy metals in Antarctic snow from Coats Land since the 1920s", Science of the Total Environment, The, 20021202 Publication	<1 %
18	c2learn.com Internet Source	<1 %
19	epdf.pub Internet Source	<1 %
20	Gang Wang, Jin-xing Hao, Jian Ma, Li-hua Huang. "chapter 64 Empirical Evaluation of Ensemble Learning for Credit Scoring", IGI Global, 2012 Publication	<1 %
21	lib.unisayogya.ac.id Internet Source	<1 %

22	moam.info Internet Source	<1%
23	dokumen.pub Internet Source	<1%
24	A.M. Strauss, G.E. Cook, Z. Bingul. "Application of fuzzy logic to spatial thermal control in fusion welding", IEEE Transactions on Industry Applications, 2000 Publication	<1%
25	C. Clifton. "Privacy-Preserving Data Mining", Studies in Fuzziness and Soft Computing, 2005 Publication	<1%
26	cimic.rutgers.edu Internet Source	<1%
27	ueaeprints.uea.ac.uk Internet Source	<1%
28	www.cs.umbc.edu Internet Source	<1%
29	"Advances in Knowledge Discovery and Data Mining", Springer Science and Business Media LLC, 2007 Publication	<1%
30	Chin-Teng Lin. "Satellite Image Classification Using Cascaded Architecture of Neural Fuzzy	<1%

Network", Advanced Information and Knowledge Processing, 2005
Publication

31	www.coursehero.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		