Distributed Privacy Preserving Classification Based on Local Cluster Identifiers

Nico Schlitter Jörg Lässig
Enterprise Application Development Group,
University of Applied Sciences Zittau/Görlitz,
Görlitz, Germany
NSchlitter@hszg.de JLaessig@hszg.de

***Abstract*—This paper addresses privacy preserving classification for vertically partitioned datasets. We present an approach based on information hiding that is similar to the basic idea of microaggregation. We use a local clustering to mask the dataset of each party and replace the original attributes by cluster identifiers. That way, the masked datasets can be integrated and used to train a classifier without further privacy restrictions. We apply our approach to four standard machine learning datasets and present the results.**

***Keywords*-Privacy Preserving Classification, Dataset Masking, Clustering**

I. Introduction

During the last decade, technological advances have significantly increased the capability of collecting and storing data. Since we are continuously confronted with data collecting processes, concerns regarding privacy issues and misusage of the collected data came up particularly with the broad usage of information and communication technologies. Especially the demand of global companies and public authorities for data analysis and data publishing concerns customers and citizens. A possible solution is provided by the research field of privacy preserving data mining, which addresses this issue and yields algorithms that ensures the privacy of provided input data during analytical processing.

Microaggregation [1] and k-anonymity [2] follow the concept of group-based anonymization and were proposed in order to protect the privacy of individuals while publishing datasets for e.g. statistical purposes. Even if a published dataset is joined with other public available datasets the anonymity of an individual is guaranteed.

Another branch of privacy preserving data mining deals with the scenario that multiple parties like to jointly analyze their datasets without revealing any private data. For that purpose, various algorithms for classification, clustering and association rule finding, which consider this specific requirement, were proposed during the last decade [3]–[5].

This paper addresses privacy preserving classification for vertically partitioned datasets. We present a new approach based on dataset masking and information hiding. The parties use a local clustering in order to mask their datasets which then can be shared between the parties in order to solve a common classification task. Since the data privacy is protected by masking of the private dataset, any traditional non privacy preserving classification algorithm can be applied to learn the characteristics of this data.

Our approach is similar to microaggregation but abstains from the descriptive manner of data aggregation as it is done by e.g. medoid aggregation. Instead, we use an identifier assigned to each cluster to mask the original data.

Please note that this work is a study of a new approach to the privacy preserving classification problem which is open for discussion and not a well-defined protocol that can be immediately implemented and used for practical applications.

The remainder of this paper is organized as follows. Section II describes the basic scenario of privacy preserving classification that is addressed by this work. Related work is discussed in Sec. III. Our new approach is presented in Sec. IV, which besides the theoretical foundation also includes a brief example in order to demonstrate its main ideas. In Sec. V we provide a security analysis of our approach, which is followed by the presentation of various experimental studies in Sec. VI. Finally, Sec. VII concludes the paper and addresses directions of future work.

II. Problem Formulation

In this work, we address the problem of privacy preserving classification in a multi-party data mining scenario. We focus on a vertically partitioned dataset which means that each party P_i holds a dataset D_p consisting of the attributes $\{A_1^p, \ldots, A_n^p\}$. In this setting of vertically partitioned data, we assume - as usual - that the distributed datasets can be joined to the integrated virtual dataset D by using the common key K for matching. Based on this integrated dataset, the parties intend to train a classifier that assigns a class label to each instance. For a certain number of records at least one of the participating parties holds additionally to its attributes also the class label, which is used to train the classifier C. In the following, we assume - without loss of generality - that party P_1 is the master party that holds the class label.

Because of privacy protection and/or legal issues, the parties are not willing to publish their datasets for data analysis purposes. Since traditional machine learning algorithms are not appropriate for this kind of task, privacy preserving

methods are necessary. Major work has been done in this research field and is briefly presented in the next section.

We propose a privacy preserving approach that does not need any further special rework for privacy purposes. It is based on information hiding and local clustering. After masking the local dataset via local cluster identifiers, the classification task can be accomplished using traditional classifiers algorithms as e.g. Decision Trees, Naive Bayes, Neural Networks or Support Vector Machines.

III. Related Work

Since privacy preserving classification over vertically partitioned data is a well studied and vivid field, only the major milestones for different machine learning algorithms are briefly mentioned in the following. In 2002, Du and Zhan proposed a privacy preserving decision tree construction method for vertically partitioned data [6]. Since the algorithm was limited to a two party case, Vaidya and Clifton [7] present a protocol in 2005 that covers the more general multi-party scenario. Using their protocol allows to construct an ID3 decision tree without revealing any private data. Thereby, the class label has to be known to just one party. The authors also address Support Vector Machines [8], [9] and propose a Naive Bayes classification that ensures the privacy of the involved data. Only when a new instance gets classified its class label gets revealed [10]. In [11], Chen and Zhong present a two party protocol for backpropagation neural network learning on vertically partitioned data and provide a correctness and security analysis of their algorithm. In [12], we propose a multi-party backpropagation protocol for privacy preserving network learning on horizontal partitioned data.

Privacy issues in dataset publishing for statistical purposes, as e.g. census data, are addressed by group-based anonymization techniques. In 1998, the concept of *k-anonymity* was proposed by Samarati and Sweeny [2], [13]. The presented technique is used to protect a dataset that is going to be published against the identification of single records by its linkage with public available datasets. To achieve this goal the values of quasi-identifier attributes of the original dataset are modified by using generalizations or suppressions. Thereby, the dataset is changed in a way that an individual found in a public available dataset can only be linked to a group of at least k records of the anonymized dataset.

Another technique for anonymizing datasets is *microaggregation*, which was proposed in [1]. In a two step process the original dataset is partitioned into several clusters of at least k records which are similar to each other. After that, each cluster gets aggregated by replacing the original attribute values of its members by a prototype of that cluster as, e.g., the medoid. In [14] the author extends the previous work and proposes methods to deal with not only numerical data but also ordinal and nominal attributes.

IV. Dataset masking using cluster identifiers

We are presenting a group based approach that is similar to microaggregation in a way that we also group data instances in order to protect data privacy. Similar to microaggregation, we propose a process of first partitioning the instances into groups (*data partition*) and second aggregating (*data aggregation*) these groups afterwards. The difference lays in the fact that we do not aggregate the group in a descriptive manner. Since, unlike to k-anonymity and microaggregation, the objective of our approach is not publishing a dataset for statistical purposes (e.g. census data), the outcoming dataset has not to be descriptive or comprehensive for human beings. Instead, we focus on masking the dataset in order to remove as much semantic information as possible while simultaneously keeping as much information as necessary to solve a subsequent classification task.

We follow the microaggregation approach, where the instances of the outcoming partition get aggregated by applying an aggregation function after partitioning the dataset. For example, a prototype that describes the group is determined (e.g. the medoid) and the original attribute values of an instance are replaced by that prototype. In contrast, we simply assign a random identifier to each group instead of trying to describe the group. Finally, we create a masked dataset by dropping the original attributes and by adding a new nominal attribute that contains only the group identifier.

A. Data Partition

The objective of data partitioning is to divide each party's dataset D_p into several disjunct groups in a way that instances within the same group are similar (with respect to some formal distance measure) to each other and dissimilar to instances of other groups. An overview considering microaggregation methods, relevant measures for disclosure risk and information loss is given in [1], [15], [16]. Since many studies have been already conducted on this subject we do not stress the different approaches of dividing a dataset into homogeneous groups in depth. Instead, we emphasize the possibility of using multiple clusterings in different subspaces of the original dataset.

Single vs. Multiple Clusterings: The simplest way for local clustering would be to use all available attributes for a *Single Clustering*. This means party P_1 uses all its attributes $\{A_1^1, \ldots, A_n^1\}$ and applies an appropriate clustering algorithm. The resulting information loss depends on the number of discovered clusters, the general characteristic of the dataset, the applied clustering technique and the similarity function.

However, not all attributes are usually used at once during the partition step of microaggregation. Instead, subsets of attributes are selected in order to partition the data separately. Each party has to decide what attributes are used

for local clustering. The process of attribute selection and how it influences disclosure risk and information loss was studied in [17]. Following this study, we suggest a *Multiple Clustering* approach which selects multiple attribute sets in order to use them for clustering. That way, information-rich subspaces can be explored in more detail in order to discover relevant groups. The usage of Single and Multiple Clustering is shown in the experiments in Sec. VI.

Partitional vs. Non-Partitional Clustering: The handling of outliers and noise is an important factor during the partition process. In the following we look into two classes of clustering methods and their impact on our approach. Using a partitional clustering algorithm (e.g. k-means [18]) guarantees that each instance is assigned to a single cluster and a specific cluster identifier. Although this approach maximizes the number of available training data it is not necessarily the best. Instances that have the characteristic of outliers get assigned to clusters anyway and will negatively influence the later classification. Non-partitional clustering algorithms (e.g. the density based DBSCAN [19]) overcome this drawback by allowing for some instances not to be part of any cluster. The instances are marked as noise and can be handled in a special manner. The simplest way of dealing with noisy instances would be to exclude them for the further classification process. If the information loss that is caused by skipping these data instances is unacceptably high we suggest to create a special noise cluster and put all noise instances in it. Applying this approach, there is still an information loss but at least there is a chance to get more information by applying another clustering based on different attributes (Multiple Clustering).

Privacy vs. Utility: Similar to k-anonymity and microaggregation the size of the clusters can be used as a measure for privacy. Following this idea, the maximal protection is reached, if all instances of the dataset D^p are member of the same cluster. Obviously, the information loss is maximized in this case as well because the attribute M_1^p of the masked dataset would have the same value for each instance and would be useless for the later classification task. At the same time, a inadequately low number of instances per cluster (in extreme one cluster for each instance) is problematic as well, because the classifier would be overfitted and not able to generalize. Consequently, the lower bound of record count per cluster is an effective control of the trade-of between privacy and utility.

B. Dataset masking

Each party masks its dataset D^p by replacing the original attributes A_i^p by the new masked attributes M_i^p. The values of M_i^p are based on the previously performed clustering. Thereby, we use the Single Clustering and the Multiple Clustering approach.

Single Clustering: After clustering the dataset by using the n available attributes $\{A_1^p, \ldots, A_n^p\}$, these original attributes are replaced by a single new attribute M_1^p whose values represent the corresponding cluster identifiers.

Multiple Clusterings: Similar to the Single Clustering, we replace the original n attributes by the masked attribute M_1^p. Further, we perform n additional clusterings using each of the original n attributes separately. Finally, we mask the data set by replacing the original n attributes by the $n+1$ new attributes $\{M_1^p, \ldots, M_n^p, M_{n+1}^p\}$.

C. Classification based on masked datasets

In the next step, the participating parties P_l that aim to solve a common classification task share their masked datasets with the master party, which then integrates the masked datasets by using the common key K. Please note, the master party has access to the class label C and can be among the parties P_l. Therefore, the master party is able to use any classification method in order to finally train the global classifier. Since the privacy protection is based on the masking of the private datasets there is no more need for additional protection mechanisms like privacy preserving decision trees [7] or support vector machines [9].

D. An Example

In the following, we demonstrate our approach by using the Iris dataset [20]. Table I shows a fraction of the original Iris dataset, which is briefly described in Sec. VI., its class labels and the four attributes. In this example we assume a two party scenario where the class label and the attributes *sepal length* (A_1^1) and *sepal width* A_2^1 belong to party P_1 and the attributes *petal length* A_1^2 and *petal width* A_2^2 belong to party P_2.

Table I
ORIGINAL IRIS DATASET

Key$K$	Class	Party 1		Party 2	
		A_1^1	A_2^1	A_1^2	A_2^2
42	setosa	4.5	2.3	1.3	0.3
43	setosa	4.4	3.2,	1.3	0.2
53	versicolor	6.9	3.1	4.9	1.5
54	versicolor	5.5	2.3	4.0	1.3
101	virginica	6.3	3.3	6.0	2.5
107	virginica	4.9	2.5	4.5	1.7

According to our *Single Clustering* approach, each party locally applies a cluster algorithm in order to assign a cluster identifier to each instance of the dataset. In our example, both parties use the partition clustering algorithm k-means ($k = 3$) to group the data within the 2-dimensional vector spaces $A_1^1 \times A_2^1$ and $A_1^2 \times A_2^2$. Each party obtains three clusters (k=3) and assigns a random identifier (M_1^1 and M_1^2) to each of them. Since cluster identifiers do not describe the meaning of clusters, they can be used to mask the dataset.

Table II shows the integration of both parties' datasets whose were masked by replacing the original attribute values with the assigned cluster identifiers.

Table II
MASKED IRIS DATASET USING SINGLE CLUSTERING

Key$K$	Class	Party 1 M_1^1	Party 2 M_1^2
42	setosa	2	1
43	setosa	2	0
53	versicolor	1	2
54	versicolor	1	1
101	virginica	0	2
107	virginica	1	1

By processing *Multiple Clusterings* it is possible to assign more than one cluster identifier to each instance of the dataset. Here we use the k-means algorithm twice in order to cluster each dimension of the two-dimensional vector space separately. The integrated masked dataset is shown in Table III.

Table III
MASKED IRIS DATASET USING MULTIPLE CLUSTERINGS

Key K	Class	Party 1		Party 2	
		M_1^1	M_2^1	M_1^2	M_2^2
42	setosa	2	2	1	1
43	setosa	2	1	1	1
53	versicolor	0	1	0	2
54	versicolor	1	2	0	2
101	virginica	1	1	2	0
107	virginica	2	2	2	0

Finally, this masked integrated dataset can be used to train a classifier with traditional machine learning algorithms as e.g. C4.5 decision trees or support vector machines.

V. SECURITY ANALYSIS

Usually, a security analysis of privacy preserving data mining methods is based on a formal privacy definition, which has to be fulfilled by a given protocol that is applied on a well specified data mining scenario. Since we suggest a new approach instead of a well formalized protocol our security analysis is less formal.

An informal but generally accepted privacy definition is as follows: A given data mining protocol that is applied on a multi-party scenario is considered privacy preserving if during processing the protocol no more information gets disclosed as it is contained in the final result anyway. This is equivalent to saying that a participating party learns nothing beyond what it knew originally and what is inherent in the final model.

In our approach, all parties share their masked dataset with the master party, which holds the class label, integrates the datasets and applies a classification algorithm in order to build the final classifier. Since the non-master parties do not receive any data, they learn nothing else but the final model. In contrast, the master party gets access to the masked dataset of all parties and learns the masked data.

Clearly, this data has to contain some information, otherwise the classification task could not be solved and the entire approach would be useless. Is there more information in the masked dataset than in the final model? Ideally yes, because a good model generalizes the dataset, which it is built on - otherwise the model would be overfitted. Following this argumentation, the presented approach obviously does not fulfill the privacy definition mentioned above.

The question is, what does the master party really learn beside cluster identifiers and how can this information be utilized? Since we use random expressions as cluster identifiers, the master party can not obtain any semantic information or meaning as it would be possible if we used aggregates like medoids instead. There is no way to bring clusters of a specific party in relation to each other since it is neither possible to estimate their location nor their distance.

Nevertheless, some information is getting known to the master party:

(i) The master party learns about the cluster sizes of the other parties and might infer a semantical meaning. For instance, a small cluster of party P2 could be an indicator for outliers.

(ii) In addition, the number of clusters is revealed and might be a useful information e.g. in the context of customer segmentation.

(iii) Furthermore, it might be possible to find relations between clusters of different parties. For instance, the appearance of frequent combinations of cluster identifiers might be found by using a correlation analysis. The master party could learn that e.g. records found in its own Cluster 1 are likely to appear in Cluster 3 of party P2. This could imply that the affected clusters are somehow related e.g. they share the same semantic meaning.

Though the original values or attributes of the private datasets will not get published the proposed approach might disclose too much information in certain scenarios. Therefore, we suggest the possibility of a small change in our approach which leads to a minimization of the disclosure risk mentioned above. Instead of assigning a single identifier to each cluster, we assign multiple identifiers and choose randomly among them during the masking step. Doing so the true number of one parties' clusters as well as their sizes are hidden because the actually identical clusters can not be linked to each other. While this approach covers the risk (i) and (ii), it also addresses risk (iii). Since the appearance probability of frequent cluster combinations decreases while the number of clusters increases, it is less likely that clusters of two parties can be linked to each other by applying a correlation analysis. Without this linkage it is consequently impossible to infer a similar meaning of two different clusters.

VI. Experiments

Since standard datasets for multi-party classification have not been published yet, we simulate this scenario by splitting standard machine learning datasets into several parts. Since we focus on vertically partitioned data, we divide the original set of attributes into 2, 3, 4 and 6 subsets in order to simulate a 2-, 3-, 4- or 6-party problem respectively.

In our experiments the local clustering of each party's dataset is done by applying the k-means algorithm [18]. Since the parameter k controls the number of clusters, we are able to study how the classification performance depends on the cluster count by varying the value of parameter k. To select the attributes that are used for the local clustering we refer to Sec. IV-A. Following the *Single Clustering* approach we apply the k-means algorithm once using all n attributes $\{A_1^p, \ldots, A_n^p\}$ of party P. The *Multiple Clusterings* case extends the *Single Clustering* case by adding n additional clusterings for each dimension A_i^p where as $i \in \{1, \ldots, n\}$.

After applying the local clustering the masked dataset is built up by assigning the corresponding cluster identifiers to each instance. Then, the masked datasets of all parties are shared with the master party which integrates all datasets by using the common key K. Finally, we apply the simple and well known C4.5 decision tree algorithm [21] to the integrated masked dataset. To verify the results and to avoid overfitting we use a 10-fold cross-validation.

All standard datasets that we used in our experiments are available at the UCI Machine Learning Repository [22]. For this work we modified the original datasets by normalizing the numerical attributes to a $[0, 1]$-interval. A brief description of the datasets as well as the outcoming results is given below. For each dataset we visualize the classification performance for the Single and the Multiple Clusterings case. The diagrams show the accuracy depending on the number of clusters (k) and different plots for the number of parties. The results of our approach are compared to a benchmark experiment which was conducted using the plain data without any privacy protection mechanisms. For this purpose the classification algorithm is applied to the unmasked integrated dataset.

A. Iris Dataset

The Iris dataset is one of the most cited datasets for pattern recognition purposes [20]. It describes the length and width of leaves of the iris plant. The dataset published in 1936 contains four numerical attributes and three classes of 50 instances each. One class is linearly separable from the other two, the latter are not linearly separable from each other.

According to Fig. 1 and Fig. 2 the classification accuracy remains quite stable while varying the number of participating parties. The number of clusters influences the performance as follows. In the case of just two clusters the inherent information loss is high and the accuracy drops below 0.8. Partitioning the local datasets in more than eleven clusters leads to a loss of generality and the consequence is a less accuracy as well. Values of k between two and eleven lead to a quite stable accuracy of about 0.95. This is even better than the benchmark results of 0.93, which were obtained from the unmasked integrated dataset.

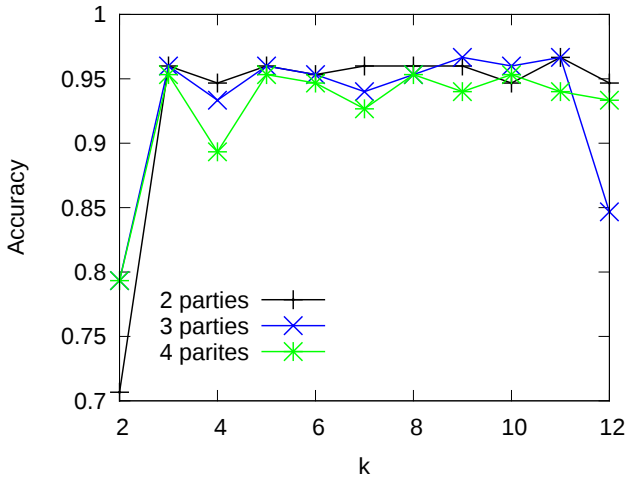

Figure 1. Plot of Iris Dataset: Single Clustering

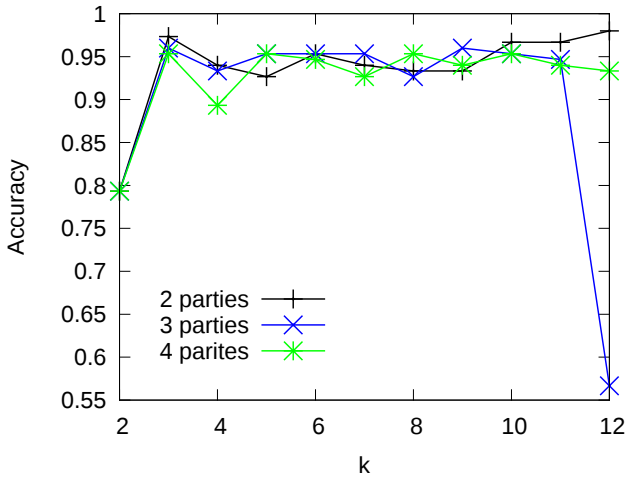

Figure 2. Plot of Iris Dataset: Multiple Clusterings

B. E.coli Dataset

The E.coli dataset was published by the Institute of Molecular and Cellular Biology at the University Osaka in 1996 [23]. It consists of 336 instances and seven numeric attributes. The class attribute can take one of seven different values and shows a skew distribution (42.6%, 22.9%, 15.5%, 10.4%, 5.9%, 1.5%, 0.6%, 0.6%).

As expected, the Multiple Clusterings case shows slightly better accuracy values than the Single Clustering. In both cases the optimal values of about 0.81 are reached for

$k = 5$. Here, the benchmark of 0.845 outperforms the privacy preserving approach slightly. As already seen at the Iris dataset, increasing k leads to a further performance loss. Thereby, the number of parties does hardly influence the classification performance.

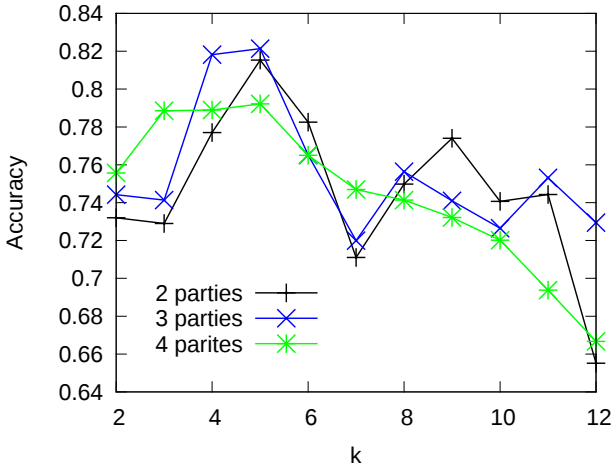

Figure 3. Plot of E.coli Dataset: Single Clustering

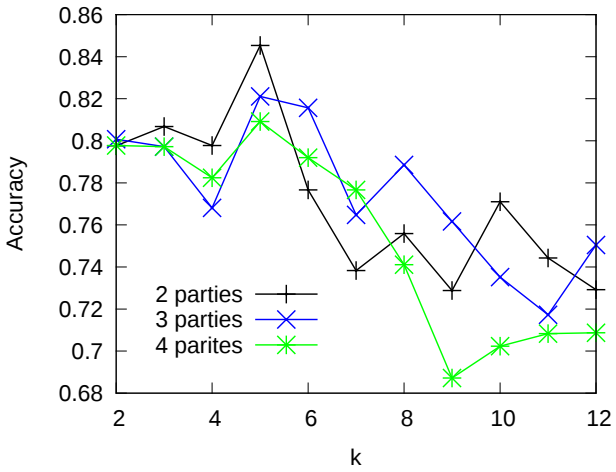

Figure 4. Plot of E.coli Dataset: Multiple Clusterings

C. Wine Dataset

The wine dataset was published in 1992 and describes results of a chemical analysis of wines grown in the same region in Italy [24]. The 178 instances are divided in three classes (33.1%, 39.9%, 27%) and consists of 13 numerical attributes.

Since the original wine dataset contains 13 attributes we were able to simulate even a 6-party scenario. However, at least in the Single Clustering case, the number of parties does not significantly influence the classification

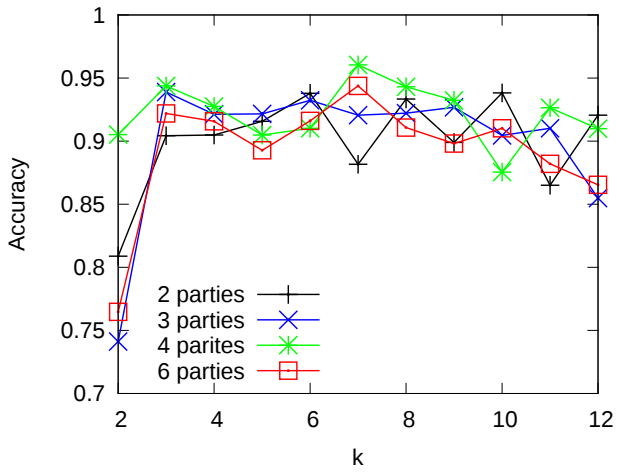

Figure 5. Plot of Wine Dataset: Single Clustering

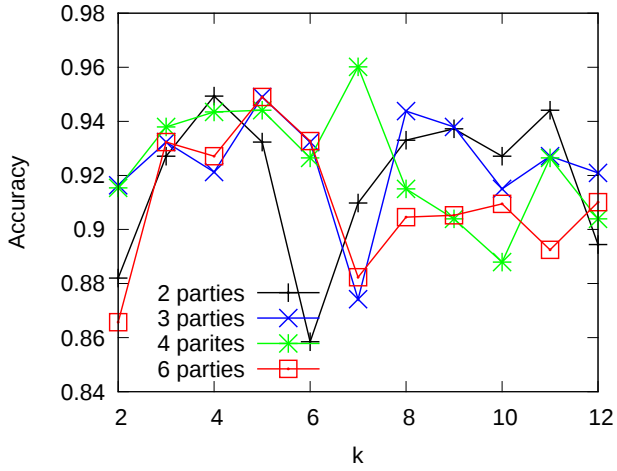

Figure 6. Plot of Wine Dataset: Multiple Clusterings

performance. The accuracy is for values of $k > 2$ quite stable at about 0.92. The multi-party case is less straight forward. For the 2-,3- and 6-party simulation, Fig. 6 shows a drop of accuracy for values of $6 \leq k \leq 7$ while the 4-party scenario reaches its optimal accuracy. Nevertheless, the achieved classifiction performance is comparable with the benchmark value of 0.93.

D. Wisconsin Breast Cancer Dataset

This breast cancer database was obtained from the University of Wisconsin Hospitals [25] and developed to a standard dataset for machine learning methods during the last decades. The original dataset consists of 699 instances and nine numerical attributes. The class attributes *malicious* (34.5%) and *benign* (65.5%) are assigned to each instance. In this work we skip the *Bare Nuclei* attribute because of missing values.

On a first glance, the Fig. 7 and Fig. 8 differ from the plots shown so far. Because of the missing downtrend while increasing k, the scaling of the accuracy axis is much more dense and the plots appear to be much more volatile. A closer look at the scaling reveals that the plots are actually similar to the ones presented before. Similar to the previous findings, the number of parties does hardly influence the performance. The average accuracy of about 0.94 equals the benchmarks that ignore privacy issues.

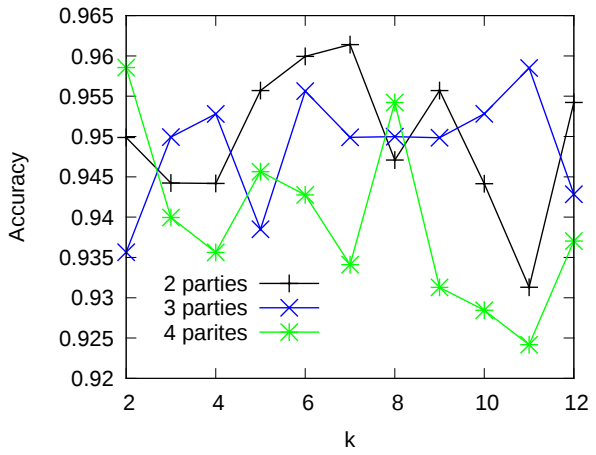

Figure 7. Plot of Breast Cancer Dataset: Single Clustering

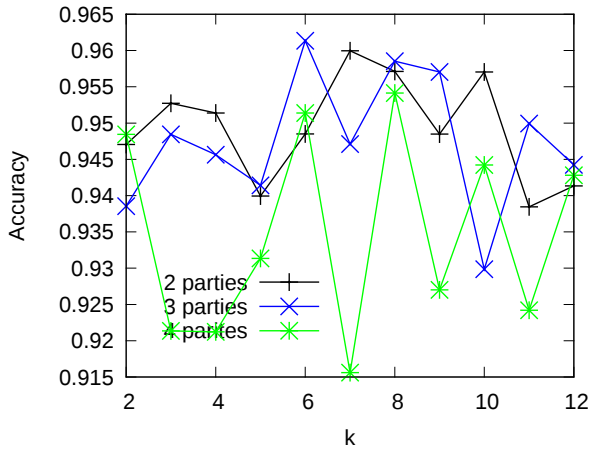

Figure 8. plot of Breast Cancer Dataset: Multiple Clusterings

E. Summarization of Results

The results of the experiments with four standard machine learning datasets show that the number of participating parties hardly influences the overall classification performance of the presented approach. The performance depends significantly on the number of clusters that the local datasets are partitioned in. This is caused either by information loss in case of a very small k or by loss of generality when k is too high. In general, values of k between three and six seem to lead to a good classification performance.

The outcoming C4.5 classification models were trained on a masked integrated dataset but reach a performance that is similar to the benchmark model which was trained on the original data. The accuracy values achieved by using the optimal k are shown in Tab. IV and V for the experiments with Single Clustering and Multi Clusterings cases respectively. As expected, the Multiple Clusterings resembles the Single Clustering case or even outperforms it due to the additional information provided by the supplemental clusters.

Table IV
BEST RESULTS SINGLE CLUSTERING

Dataset	Benchmark	Number of Parties			
		2	3	4	6
Iris	0.927	0.966	0.966	0.953	-
E.Coli	0.845	0.815	0.821	0.789	-
Wine	0.933	0.938	0.938	0.960	0.960
Cancer	0.940	0.959	0.961	0.954	-

Table V
BEST RESULTS MULTIPLE CLUSTERINGS

Dataset	Benchmark	Number of Parties			
		2	3	4	6
Iris	0.927	0.973	0.960	0.953	-
E.Coli	0.845	0.845	0.821	0.809	-
Wine	0.933	0.949	0.949	0.960	0.949
Cancer	0.940	0.959	0.961	0951	-

VII. CONCLUSION AND OUTLOOK

In this paper we proposed a new approach for privacy preserving classification on vertically partitioned datasets in a multi-party scenario. Our course of action was influenced by the concept of group-based anonymization as it is used in the k-anonymity and microaggregation approaches. A local clustering divides the original instances of a dataset into several clusters and to each cluster a unique identifier is assigned. Later on, these cluster identifiers are used to mask the original dataset by replacing its attributes. Thereby, the values of new attributes correspond to the identifiers of the discovered clusters. Subsequently the partitioning parties share their masked datasets. Since no more privacy protection is necessary, usual classification techniques can be used in order to train a classifier based on the masked integrated dataset. We applied this approach to four standard machine learning datasets and presented the results. The findings show that our approach leads to similar classification performance as it would be achieved by applying traditional non-privacy preserving machine learning algorithms. In future work we plan to study the impact of different clustering algorithms on the final classification performance. Especially the density-based DBScan but also hierarchical clustering algorithms seem to be worth further investigations.

ACKNOWLEDGMENT

Zaigham Faraz Siddiqui implemented an early stage of the RapidMiner-Plug-In which was used for this work. The computational power for the related experiments was obtained from the *distributedDataMining project* (http://www.distributedDataMining.org) that is based on the distributed grid computing platform BOINC [26].

REFERENCES

[1] J. Domingo-Ferrer and V. c Torra, *A quantitative comparison of disclosure control methods for microdata*. Elsevier, 2001, pp. 111–133.

[2] L. S. Pierangela Samarati, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," in *IEEE Symposium on Research in Security and Privacy*, 1998.

[3] J. Vaidya, C. Clifton, and M. Zhu, *Privacy-Preserving Data Mining*, 1st ed., ser. Advances in Information Security. Springer-Verlag, 2005.

[4] L. Wang, S. Jajodia, and D. Wijesekera, *Preserving Privacy in On-Line Analytical Processing (OLAP) (Advances in Information Security)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[5] C. C. Aggarwal and P. S. Yu, *Privacy-Preserving Data Mining: Models and Algorithms*. Springer Publishing Company, Incorporated, 2008.

[6] W. Du and Z. Zhan, "Building decision tree classifier on private data," in *Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14*, ser. CRPIT '14. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2002, pp. 1–8.

[7] J. Vaidya and C. Clifton, "Privacy-preserving decision trees over vertically partitioned data," in *The 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*. Storrs, Connecticut: Springer, 08 2005.

[8] H. Yu, J. Vaidya, and X. Jiang, "Privacy-preserving svm classification on vertically partitioned data," in *PAKDD*, ser. Lecture Notes in Computer Science, W. K. Ng, M. Kitsuregawa, J. Li, and K. Chang, Eds., vol. 3918. Springer, 2006, pp. 647–656.

[9] J. Vaidya, H. Yu, and X. Jiang, "Privacy-preserving svm classification," *Knowl. Inf. Syst.*, vol. 14, pp. 161–178, January 2008.

[10] J. Vaidya and C. Clifton, "Privacy preserving naïve bayes classifier for vertically partitioned data," in *SDM*, M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, Eds. SIAM, 2004.

[11] T. Chen and S. Zhong, "Privacy-preserving backpropagation neural network learning," *Trans. Neur. Netw.*, vol. 20, pp. 1554–1564, October 2009.

[12] N. Schlitter, "A protocol for privacy preserving neural network learning on horizontal partitioned data," in *Privacy Statistics in Databases (PSD) 2008*, Istanbul,Turkey, September 2008, p. on CD.

[13] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, October 2002.

[14] V. Torra, "Microaggregation for categorical variables: A median based approach." in *Privacy in Statistical Databases*, ser. Lecture Notes in Computer Science, J. Domingo-Ferrer and V. Torra, Eds., vol. 3050. Springer, 2004, pp. 162–174.

[15] E. Fayyoumi and B. J. Oommen, "A survey on statistical disclosure control and micro-aggregation techniques for secure statistical databases," *Softw. Pract. Exper.*, vol. 40, pp. 1161–1188, November 2010.

[16] V. c Torra and J. Domingo-Ferrer, *Disclosure control methods and information loss for microdata*. Elsevier, 2001, pp. 91–110.

[17] J. Nin, J. Herranz, and V. Torra, "How to group attributes in multivariate microaggregation." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. Supplement-1, pp. 121–138, 2008.

[18] E. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–780, 1965.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp. 226–231.

[20] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[21] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[22] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[23] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins," in *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1996, pp. 109–115. [Online]. Available: http://portal.acm.org/citation.cfm?id=645631.662879

[24] S. Aeberhard, D. Coomans, and O. de Vel, "Comparison of Classifiers in High Dimensional Settings," Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep. 92-02, 1992.

[25] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proceedings of The National Academy of Sciences*, vol. 87, pp. 9193–9196, 1990.

[26] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, ser. GRID '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 4–10.