Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.020164
Article

Machine Learning Privacy Aware Anonymization Using MapReduce Based Neural Network

U. Selvi* **and S. Pushpa**

Department of Computer Science and Engineering, St. Peter's Institute of Higher Education and Research, Chennai, India
*Corresponding Author: U. Selvi. Email: slvunnikrishnan@gmail.com
Received: 12 May 2021; Accepted: 22 June 2021

Abstract: Due to the recent advancement in technologies, a huge amount of data is generated where individual private information needs to be preserved. A proper Anonymization algorithm with increased Data utility is required to protect individual privacy. However, preserving privacy of individuals whileprocessing huge amount of data is a challenging task, as the data contains certain sensitive information. Moreover, scalability issue in handling a large dataset is found in using existing framework. Many an Anonymization algorithm for Big Data have been developed and under research. We propose a method of applying Machine Learning techniques to protect and preserve the personal identities of Individuals in BigData framework, which is termed as BigData Privacy Aware Machine Learning. For addressing a large volume of data, MapReduce-based neural networks parallelism is taken into consideration with classification of data volume. Human contextual information as applied through collaborative Machine Learning is proposed. The result of our experiment shows that relating human knowledge to neural network and parallelism by MapReduce framework can yield a better and measurable classification results for large scale Applications.

Keywords: Privacy aware machine learning; anonymization; k-anonymity; bigdata; mapreduce; back-propagation neural network; machine learning

1 Introduction

BigData as the name implies refers to massive amount of data, generated at high speed with different data types that cannot be processed by tools available for Database Management and are problematic to capture, collect, explore, share, examine and visualize data [1].

BigData and 4v's characteristics:

i) Volume states the large quantity of Data Generation and Data Collection;
ii) Velocity states the timeliness of the data received at data pool for analysis;
iii) Variety states data forms like unstructured, structured and semi-structured data; and
iv) Value states information concealed from data.

MapReduce, the customary computation model is used for handling BigData applications [1,2]. It is a framework used for processing larger datasets and is consistent, fault-tolerant, accessible, and self-balancing when the size of dataset increases. MapReduce framework [3] handles the larger sets using Map and Reduce Functions. Basically, a map functions produces 〈key, value〉 pairs as intermediate results by processing data. A reduce function sorts and merges the 〈key, value〉 pairs collected from multiple mappers and applies the results for secondary processing. Finally the reducer generates the results, based on the input which is collected from the outputs of multiple mappers.

Data Anonymization is the technique of protecting sensitive information from disclosure and preserving the privacy of the users of the application. In our paper, the standard k-Anonymity algorithm is chosen to preserve privacy.

Artificial Neural Networks [4] (ANNs) is capable of modelling & processing non-linear relationships between input and output in parallel and have been widely used in various research scenario. One of the implementation of ANN is neural network which uses Back-Propagation (BPNN) approach, which is evidenced to be efficient in terms of approximation capability. An 'n' number of hidden inputs and outputs layers are found in BPNN, with each layer containing neurons. BPNN uses an error-back propagation mechanism for training data and employs feed forward network to produce the required output.

This paper tries to focus on MapReduce approach which implements Back-Propagation Neural Network (MRBPNN) by considering Classification of data Volume i.e., we aim to implement Machine Learning Algorithm in MapReduce for parallel processing.

Malle et al. [5] Aims to provide the report on information loss and Quasi-identifier distributions by defining a k-factor proposed by user. But the algorithm is found to be not interactive and doesn't achieve the expected results during the learning phase. The user has to check whether the expected result is achieved or not and decide after completion of Anonymization run upon using Cornell Anonymization Toolkit (Cat) [6]. Our methodology adjusts algorithmic factors upon each (batch of) manual interruptions, to make the algorithm to be adapted in real-time.

Xu et al. [7] Proposes to construct generalization hierarchies by allowing human interventions to set constraints on attributes in the process of anonymization.

Our Contributions

We propose Hadoop MapReduce framework that implements Back-propagation Neural Network to achieve k-anonymity for large scale application.

Our contribution is summarized as follows:

i) Fast Correlation based on Feature Selection Algorithm using Map Reduce is used for pre-processing the dataset to select relevant feature.
ii) Then, the pre-processed data is fed to the MapReduce Framework. Each mapper has a Back Propagation Neural Network which maps the data to form equivalent groups which form clusters. The algorithm begins by selecting the next candidate for merging until the cluster reaches the size k. The k-anonymity criterion is satisfied by combining all data points to form clusters for a given dataset. The intermediate result from the mapper is fed as input to reducer function.
iii) BPNN uses the error propagation method to tune the network parameter until it satisfies the k-anonymity.

The paper organized as follows:

Section 2: Delivers contextual evidence around MapReduce-based Back Propagation Neural network and the architecture of BPNN. Section 3: Explains k-anonymity and Section 4: Deals with BigData

MapReduce. Section 5: Describes the Iterative Machine learning to achieve Anonymity. Section 6: Deliberation and Analysis of the empirical studies. Section 7: Summarizes the conclusion of the paper.

2 Paralleling Neural Network

This section explains the Paralleling Neural Networks with Back-Propagation.

2.1 Back-Propagation Neural Networks (BPNN)

Propagating errors in backward direction for training data is Back-Propagation Neural Network. It is a multi-layered structure and feeds forward network. Input-output mappings using BPNN can be performed with a large volume of data, without having adequate knowledge on mathematical equation involved. BPNN tunes the network parameter to achieve k-anonymity in the process of the error propagation. Fig. 1 depicts the BPNN which has a number of inputs and outputs with a multi-layered network structure. A BPNN has three explicit layers: (i) the input, (ii) the output and (iii) the hidden layers. It is the commonly accepted network structure to fit a mathematical equation and to map the relationships between inputs and outputs [8].

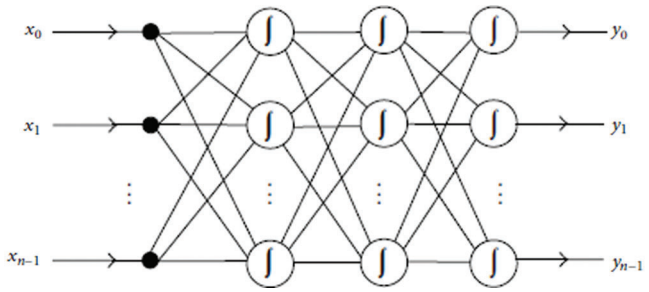

Figure 1: Neural network with back-propagation

2.2 The Design of MapReduce Back-Propagation Neural Network

Consider a testing instance j = $\{b_1, b_2, b_3, \ldots, b_{jk}\}$, $q_j \in$ Q, where

i) Data instance is denoted by q_j ;
ii) Dataset is denoted by Q;
iii) The dimension of q_j, the input quantity of neural network, denoted by jk;
iv) The inputs are represented as $\langle instance_n, target_n, \text{type}\rangle$;
v) Neural Network (NN) input $instance_n$ signified by q_j;
vi) $target_n$, signifies the anticipated yield, if $instance_n$ is a training instance;
vii) Two values of field type includes: 'trainset' and 'testset', which are marked based on the category of $instance_n$; if 'testset' value is fixed, $target_n$ field is shown blank.

Initially, records covering instances are kept into HDFS. Each record has training and testing instances. Consequently, the record number η specifies the quantity of mappers used. The data chunk or the training data is fed as input to each mapper. Fig. 2 shows the architecture of MapReduce-based Back Propagation Neural Network (BPNN).

Initializing a neural network with each mapper function is the first step where the Algorithm begins. As a consequence, in a cluster, there are ð neural networks having exactly the same structure and parameters. As the training data is fed into the Mapper, each mapper reads data and picks a first cluster randomly or pre-defined from the data row. Then, the process continues in selecting the finest candidates for integration by

reducing GIL and proceeds to reach the cluster size of k. When the cluster size reaches size of k, the next cluster with new data point is chosen as initiator; the above said process is repeated to form multiple clusters from the data points, to satisfy k-anonymity for the specified number of dataset. The Error Propagation concept in neural network helps in maintainining a minimum Information Loss using the GIL measure.

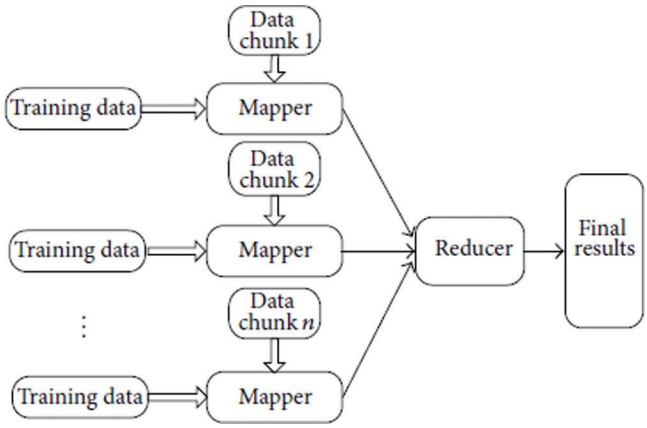

Figure 2: Map Reduce based BPNN

Finally Reducer maintains a final output based on the data collected from Mapper. Thus MapReduce solves the scalability problem of BigData and interactive Machine Learning in Neural network satisfies k-anonymity.

Algorithm: Anonymity Algorithm with Back-Propagation Neural Network in MapReduce

Input: U,Q; **Output**: A ; m number of mappers ; one reducer
|cl| designates the cluster cardinal; size([i1, i2]) specifies interval size (i2 − i1);
$\wedge(\omega)$, $\omega \in H_{C_j}$ is the sub-hierarchy of H_{C_j} embedded in ω;
$height(H_{C_j})$ signifies the altitude of the tree hierarchy H_{C_j}

i) BP-NN was constructed in each mapper with inp inputs, θ outputs, n neurons in hidden

ii) Initially, ω_{uv}= $random_{1uv} \in$ (-1,1), $\varnothing_v$=$random_{2v} \in$ (-1,1)

iii) $\forall r \in R$, r_u={ $b_1, b_2, b_3, \ldots..b_{inp}$}
Input $b_i \rightarrow inp_i$, $neuron_j$ in hidden layer computes

$$I_{vn} = \sum_{u=1}^{inp} b_u \,.\, \omega_{uv} + \varnothing_v$$

$$\theta_{vn} = \frac{1}{1 + e^{I_{vn}}}$$

iv) Input $\theta_v \rightarrow Out_u$, $neuron_v$ in output layer computes

$$I_{v0} = \sum_{u=1}^{n} \theta_{un}.\ \omega_{uv} + \varnothing_v$$

$$\theta_{vo} = \frac{1}{1 + e^{I_{vn}}}$$

v) In each Output, compute
Err_{vo}= $\theta_{vo}(1-\theta_{vo})(target_v - \theta_{vo})$

(Continued)

Algorithm (continued).

vi) In every hidden layer, compute
$Err_{vn}= \theta_{vn}(1- \theta_{vn}) \sum_{u=1}^{o} Err_u \omega_{uo}$

vii) Update
$\omega_{uv}= \omega_{uv}+\eta. Err_v.$
$\theta_v \varnothing_v = \varnothing_v + \eta. Err_v$; Repeat (iii), (iv), (v), (vi), (vii)
Until min(E[e^2] = min (E [$(target_v - \theta_{vo})^2$]) and k-anonymity satisfied with minimum Information Loss using the formula Generic Information Loss(GIL):

$$GIL(cl) = |cl|. \left(\sum_{j=1}^{s} \frac{size(gen(cl)[Nj\ ])}{size(minx_N(X[Nj\ ]), maxx\_N\ (X[Nj\ ]))} + \sum_{j=1}^{t} \frac{height(\wedge(gen(cl)[Nj\ ])}{height(H_{C_j})}\right.$$

Training terminates

viii) Divide Q into $\{Q_1, Q_2, Q_3 \ldots..,Q_m\}$, $U_{u=0}^{m} Q_u$=Q

ix) Each mapper inputs $q_v = \{ b_1, b_2, b_3, \ldots..b_{inp}\}$, $q_v \in Q_u$

x) Execute step (iii), (iv)

xi) Mapper yields the result as $<q_v, \theta_v>$

xii) Reducer assembles and combines all $<q_v, \theta_v>$

xiii) Reiterate (ix), (x), (xi), (xii); Till Q is navigated.

xiv) Reduce function yields A
Process stop.

3 Data Anonymization

Data Anonymization [9] is the process of screening identity and sensitive records for the possessors of information records. The main objective of Data Anonymization [10] is to preserve privacy by exposing aggregate information to data scientist for analytics and mining. Data given for data mining and analysis should conserve a proper stability among the utility and privacy. The bench mark algorithm for Anonymization is k-anonymity which uses the concept of Generalization of attributes and Suppression of tuples (i.e., Suppression of attributes).

K-anonymity

The Classification of Dataset includes Personal identifiers, Sensitive information and Quasi-identifiers

- A Personal identifier is an attributes which straightaway recognizes an individual without any further analysis or cross-reference. Examples are individual mail-ID or Social Security Number (SSN). This category of data is usually dangerous as it reveals the individual identity and needs to be removed.
- Sensitive information is the vital information that can be used for research purposes and for mining. Examples include disease classification, medication intaken information or salary information of an individual. Data relating to these are required for analysis and should be conserved as an anonymized dataset. So, such data cannot undergo generalization or suppression.
- Other types of attributes include Quasi-identifiers (QI's), which don't directly recognize the individual. However, when aggregated information is given individuals can be reconstructed from them.

For illustration, report in 2002 said that individual information in a certain region can be revealed via attributes with zip code, gender and birth date. Based on this information, it can be inferred that,

Quasi-Identifier comprise dynamic information for analysis of research applications and [11] they can be generalized or suppressed based on conciliation between privacy which inhibits information loss and data utilization.

In k-anonymity [12], the information of an individual cannot be revealed by exposed information and revealed information have a minimum of k − 1 people with the same information in the cluster. At least, k-record should have same Quasi-identifier. For instance, in a released table, Birth date of a person and gender attribute are the QID, to achieve k-anonymity, k-people should have the same date of birth and gender in the given datasets. In a *k*-anonymous table, there is no unique record and k-1 record has similar QID values. Generalization and suppression are the key concepts to achieve k-anonymity. To anonymize a data structure, it uses an algorithm given the General Information Loss (GIL) through Anonymization. Generally General Information Loss (GIL) is defined as amount of information loss occurring through generalization of attributes as in Eq. (1).

General Information loss (GIL) is denoted as:

$$\text{GIL (cl)} = |\text{cl}|.\sum_{j=1}^{s}\frac{\text{size(gen(cl)[Nj])}}{\text{size}(\text{minx}_{\text{N}}(\text{X[Nj]}),\text{maxx}_{\text{N}}(\text{X[Nj]}))} + \sum_{j=1}^{t}\frac{\text{height}(\wedge(\text{gen(cl)[Nj]})}{height\left(H_{C_j}\right)} \quad (1)$$

where:

- |cl| signifies the cluster cl's cardinal;
- size([i1, i2]) signifies the size of the interval (i2 − i1);
- $\wedge$ (ω), $\omega \in H_{C_j}$ signifies sub-hierarchy of H_{C_j} embedded in ω;
- $height\left(H_{C_j}\right)$ signifies the altitude of the tree Hierarchy H_{C_j} ;

The training data is fed into a mapper; each mapper reads the data and picks a first cluster randomly or as pre-defined from the data row. Then, the process continues in selecting the finest attributes for integration by reducing General Information Loss (GIL) and the process repeats itself to retain the cluster size of k. When the cluster size reaches the size of k, the next cluster with fresh data argument is elected as the originator; for the given dataset, this procedure iterates until the entire data arguments are combined to form new clusters to satisfy anonymity algorithm.

k-anonymous [13] datasets with reference to Quasi-identifier has the size of the similarity class to be 'k' or more with reference to the Quasi-identifier. Data Utility for Classification is not considered by Generalization and Suppression after anonymity. Well ahead of the k-anonymity there should be l-diversity [14] (where every cluster should maintain l-diverse sensitive data), t-closeness (with a threshold of t, native dissemination over sensitive data must not deviate from its comprehensive dissemination), m-variance, differential privacy (a noise is injected into the dataset for securely releasing sensitive information) which is proposed.

4 BigData: MapReduce, Hadoop

This section explains the two main concepts for BigData processing. The following sub-sections focus on the MapReduce programming model, pre-processing data and FCBF in map-reduce framework.

4.1 The MapReduce: Computing Model

In the revolution of BigData, MapReduce [15] is the customary computing model for processing large volumes of dataset using a cluster of commodity computers. Hadoop [16,17] an open source, is the popular implementations of MapReduce model.

HDFS which is used for data Management and MapReduce [18] are the two main concepts in Hadoop framework. For running jobs and to process data in Hadoop cluster, it has a Namenode and Datanode. The namenode is accountable for the cluster's metadata and the Datanode is the actual processing node which has Map and Reduce functions. The data fed as input is split into a number of small chunks of equal size, while submitting job to Hadoop and is stored in the HDFS. To preserve data reliability, every data portion of the data can have one or additional copies as per Hadoop cluster configuration. Mappers duplicate and deliver data based on data vicinity. Finally, HDFS has the concluding response which is organized, combined and produced by reducers.

To process and handle a large amount datasets in parallel, scalable and fault-tolerant MapReduce framework [19] for data processing is developed. MapReduce has a two-fold basic task: Map and Reduce functions. A mapper, based on the input key-value pair, yields an intermediary key-value pair. Each Mapper has a Neural Network based on Back propagation Algorithm. Each intermediary key-value pair is combined based on the key and communicated to the Reducer, which compresses the values to make them reduced.

4.2 Pre-Processing Data

Based on our previous work [20], we elaborated the measure of goodness of feature for classification. In terms of Correlation Analysis [21], a worthy feature greatly related to the class is nevertheless not associated with any further features.

The Fast Correlation Feature Selection [22] algorithm explores search space by using the best-first search Algorithm. The algorithm begins with a void list of features and upon iteration of search; all probable sole feature expansions are produced. The novel subsets are estimated and added to a precedence queue permitting to Improvement. In the consequent iteration, the optimum subset of the queue is designated for improvement in the same way as the first certain void subgroup. The next finest subset is designated from the queue, if the subsequent finest subset fails to produce enhancement. After five successive failure (since it is the criterion for stopping), the FCFS algorithm stops altogether.

The last CFS [23] element is a discretionary step. The FCFS algorithm selects feature subsets with low redundancy and high correlation within the class. However, in certain cases, additional features that are nearly predictive in a minor area of the instance space may occur that can be leveraged by some classifiers. To take these features in the subgroup subsequent to the search, the FCFS can use an experimental study that permits the presence of all features. Hence the association of features within the class is more sophisticated than the correlation between the features already selected.

4.3 FCBF in MapReduce Framework

As shown in Fig. 3 a huge amount of data collected is allocated as chunk and each chuck is fed to the Mapper. At each Mapper, a Fast-Correlation Based Features Selection algorithm is provided to select the optimum subset of features and to remove the duplicate feature subset with increased data utility. Finally, the k-anonymity algorithm in reducer confirms the privacy of an individual. Henceforth, the outcome of Map-Reduce is Anonymized Dataset which satisfies k-anonymity. Upon applying FCFS algorithm to the large Dataset, it is found that execution time decreases with the data volume as shown in Fig. 4. Hence for Pre-processing process, FCFS algorithm is selected. This pre-processed data is fed as input to the next stage MapReduce for anonymization process.

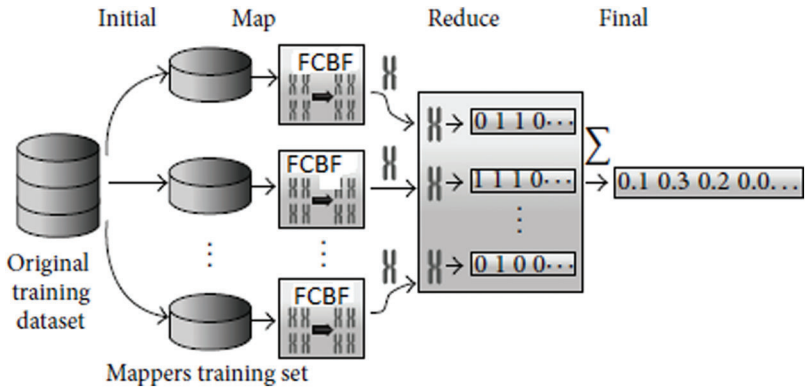

Figure 3: K-anonymity based FCBF in Map-Reduce

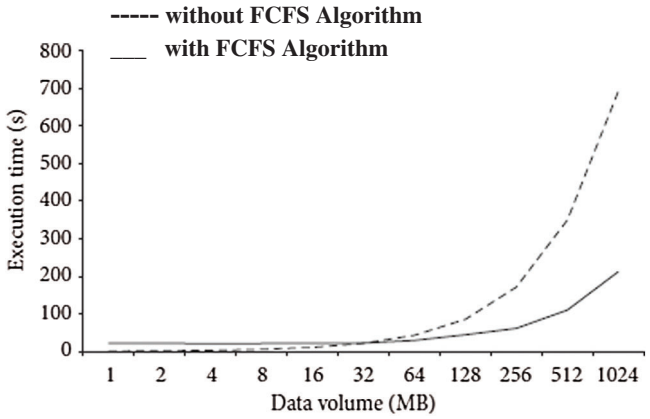

Figure 4: Execution Time decreases as data volume increases by the use of FCFS Algorithm

5 Interactive Machine Learning

Interactive Machine learning algorithms [24] infer a positive or negative reinforcement by the way of human interaction of outward oracle with their inner working mechanism [25]. Our methodology alters the algorithmic factors upon each consignment of manual interruptions, permitting them to adjust an approach of including human decisions in real-time applications [26].

This is done into the process of anonymization by permitting them to fix limitations on instance generalization; besides, generalization, hierarchies were constructed comprising domain-specific ontologies.

6 Performance Evaluation

The parallel BPNNs using Hadoop with MapReduce computing model was implemented to test our work. Multi-Variate Adult Dataset is used to implement our proposed work. It has both Categorical and Integer Attribute (14 attributes), 48842 instances having missing values. The exactness of the algorithms was calculated by changing the value of k starting from 10 and further increasing its value upto 1000. The computation efficiency was estimated by varying the datasets size from 1MB to 1GB.

A four different Classification algorithm was applied along with iML [27,28] in Neural Network for large scale applications on three target attributes to generate anonymized datasets.

The following processing pipeline design is applied as follows:

i) The original datasets are pre-processed using Fast Correlational Feature selection algorithm, followed by the application of k-anonymity algorithm with the value of k ranging from 5 to 100 as [5,10,20,50,100,200] and 129 completely dissimilar weight classification (iML, bias, equal) to create anonymized datasets.

ii) We attempted to execute four classification algorithms on all of the datasets and relevant F1 score were compared; the rationale behind choosing numerous algorithms was to discover if anonymization would produce completely different performances on multiple mathematical methodologies for classification. The different algorithms used were linear support vector machines, logistic regression, gradient boosting (ensemble, boosting) equally as Random Forest (ensemble, bagging). Considering classification goal as education, fourteen completely diverse education levels exist on adult dataset. Broadly it can be classified into four classes as 'advanced studies', '<=bachelors', 'high school' and 'pre high school'.

iii) For every group of classification goal (education, marital status, income) and weight classification (iML, bias, equal) we averaged the corresponding outcomes. Outcomes stay designed for each goal (target), such as these permit improved evaluation among diverse classifiers. Figs. 5–7 show the result on applying different classifier based on attributes selected for anonymization.

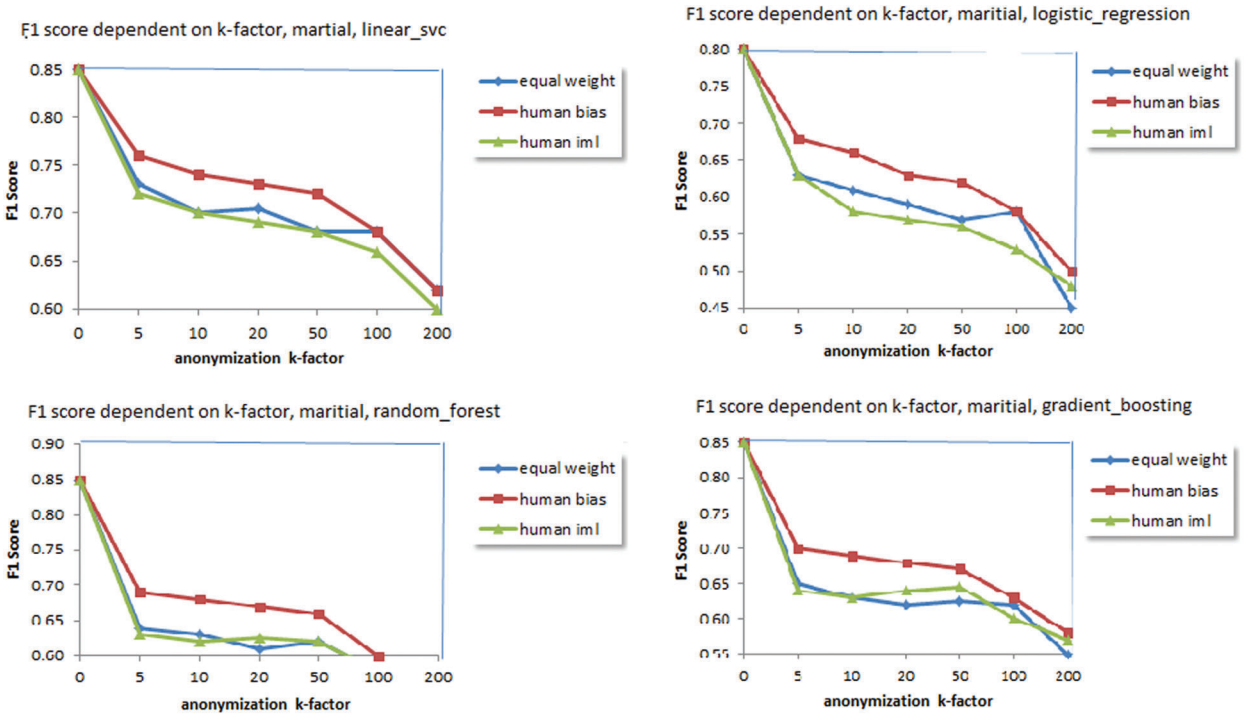

Figure 5: Human bias overtaking both equal weights and human interaction parameters when marital status is taken as the target attributes

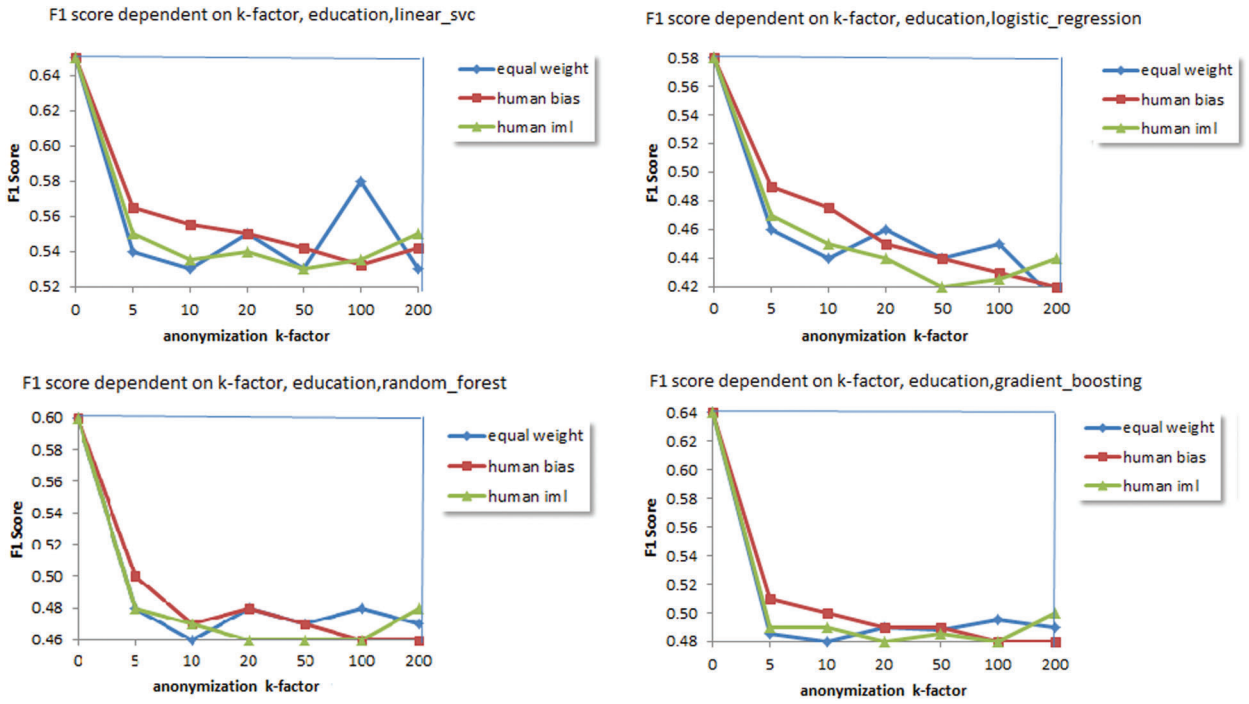

Figure 6: Human bias performs marginally well in comparison with equal weights / iML parameters with different values of k, nonetheless not subsequently as already stated, when education is taken as the target attribute

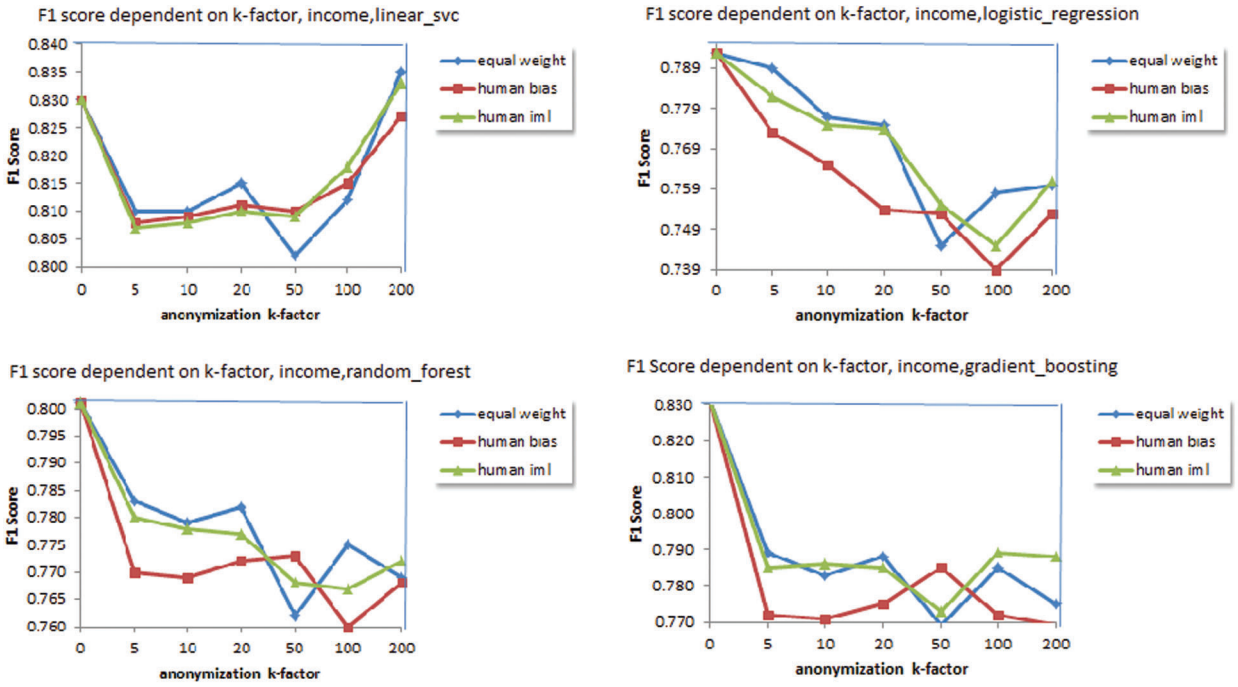

Figure 7: iML-based results generally outperform bias excluding linear SVC; However, they are incompetent of overtaking the firmly equal setting when income is taken as the target attribute

7 Conclusion

We have outlined parallel neural networks using manual interruptions to tolerate on the mission of Anonymization through iML. This is built on Hadoop MapReduce programming structure in associations with classification datasets. We devised an experiment concerning clustering of data arguments probably with manual inclination for conservation of attributes and the resultant constraints on classification of anonymized data into classes of income, education and marital status are verified. The outcomes demonstrate that human bias with MapReduce in Neural network can positively contribute to ordinary presentation areas, whereas supplementary difficult applications require trained professionals or better data preparation. Further research is required for privacy preservation when the data needs to be analyzed, shared and mined. This work can be extended as a Deep Learning based neural network to achieve k-anonymity for Large Data Applications in our subsequent work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Garcia, S. R. Gallego, J. Luengo, J. M. Benitez and F. Herrera, "Big data pre-processing: Methods and prospects"," *Big Data Analytics*, vol. 42, no. 6, pp. 1911–1920, 2019.

[2] M. chen, S. Mao and Y. Lin, "Big Data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[3] B. He, W. Fang, Q. Luo, N. K. Govindaraju and T. Wang, "Mars: A map-reduce framework on graphics processors," in *Proc. of the 17th Int. Conf. on Parallel Architectures Computational Intelligence and Neuroscience and Compilation Techniques (PACT '08)*, Toronto Ontario Canada, pp. 260–269, 2008.

[4] Y. Liu, J. Yang, Y. Huang, L. Xu and S. Lie, "Map-reduce based parallel neural network enabling large scale machine learning," *Hindawi*, vol. 2015, pp. 1–13, 2015.

[5] B. Malle, P. Kieseberg and A. Holzinger, "Interactive anonymization for privacy aware machine learning," in *European Conf. on Machine Learning and Knowledge Discovery ECML-PKDD*, Skopje, Macedonia, The Former Yugoslav Republic of Macedonia, pp. 15–26, 2017.

[6] X. Xiao, G. Wang and J. Gehrke, "Interactive anonymization of sensitive data," in *Proc. of the 35th SIGMOD Int. Conf. on Management of Data - SIGMOD '09*, Association for Computing Machinery, New York, pp. 1051–1054, 2009.

[7] K. Xu, H. Yue, L. Guo, Y. Guo and Y. Fang, "Privacy-preserving machine learning algorithms for big data Systems," in *Int. Conf. on Distributed Computing Systems*, IEEE, Columbus, USA, 2015.

[8] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Privacy, Security, and Trust in KDD*. vol. 5456, United States: Springer, pp. 33–54, 2009.

[9] L. Zheng, H. Yue, X. Pan, M. Wu and F. Yang, "K-anonymity location privacy algorithm based on clustering," *IEEE Access*, vol. 6, pp. 28328–28338, 2018.

[10] Y. Miche, W. Ren, L. Oliver and S. Holtmanns, "A framework for privacy quantification: measuring the impact of privacy techniques through mutual information, distance mapping, and machine learning," *Springer Cognitive Computation*, vol. 11, no. 2, pp. 241–261, 2019.

[11] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2012.

[12] K. L. Du, "Clustering: A neural network approach," *Neural Networks Elsevier*, vol. 23, no. 1, pp. 89–107, 2010.

[13] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 47–77, 2011.

[14] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2012.

[15] C. L. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, no. 4, pp. 314–347, 2014.

[16] Apache Hadoop 2015. [Online]. Available: http://hadoop.apache.org.

[17] J. Venner, *Pro Hadoop*. New York, NY, USA: Springer, 2009.

[18] X. Zhang, C. Yang, S. Nepal, C. Liu, W. Dou *et al., A map-reduce based approach of scalable multidimentional anonymization for big data privacy preservation on cloud*. IEEE Third International Conference on Cloud and Green Computing, USA, 2013.

[19] U. Selvi and S. Pushpa, "A review of big data and anonymization algorithms," *International Journal of Applied Engineering Research*, vol. 10, no. 17, pp. 13125–13130, 2015.

[20] U. Selvi and S. Pushpa, "Big data feature selection to achieve anonymization, Invention Communication and computational technologies," *Spinger*, vol. 637, pp. 59–67, 2020.

[21] B. Zhang, N. Mohammed, V. Dave and M. A. Hasan, "Feature selection for classification under anonymity constraint," *ACM*, vol. 10, no. 1, pp. 61–81, 2015.

[22] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. of the Twentieth Int. Conf. on Machine Learning*, Washington, DC, 2003.

[23] D. Peralta, S. Del Rio and S. Ramirez-Gallego, "Evolutionary feature selection for big data classification: A map reduce approach," *Hindawi*, vol. 2015, pp. 1–12, 2015.

[24] B. Malle, P. Kieseberg and A. Holzinger, "Do not disturb? classifier behavior on perturbed datasets," in *Machine Learning and Knowledge Extraction, IFIP CD-MAKE*, *Lecture Notes in Computer Science LNCS*, vol. 10410. Cham: Springer, pp. 155–173, 2017.

[25] A. Holzinger, "Interactive machine learning for health informatics: When do we need the human-in-the-loop?," *Springer Brain Informatics (BRIN)*, vol. 3, no. 2, pp. 119–131, 2016.

[26] C. Moque, A. Pomares and R. Gonzalez, "A proposal for interactive anonymization of electronic medical records," *Procedia Technology*, vol. 5, pp. 743–752, 2012.

[27] B. Malle, P. Kieseberg, E. Weippl and A. Holzinger, "The right to be forgotten: towards machine learning on perturbed knowledge bases," in *Int. Conf. on Availability, Reliability, and Security*, Springer, Salzburg, Austria, pp. 251–266, 2016.

[28] A. Holzinger, M. Plass, K. Holzinger, G. C. Crisan, C. M. Pintea *et al.,* "Towards interactive machine learning (iml): Applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach," in *IFIP Int. Cross Domain Conf. and Workshop (CD-ARES)*, Heidelberg, Berlin, New York, Springer, pp. 81–95, 2016.