Differentially Private Learning from Label Proportions*

Timon Sachweh[0000−0002−0347−5760], Daniel Boiar[0000−0002−6856−9848], and Thomas Liebig[0000−0002−9841−1101]

TU Dortmund University
Dortmund, Germany
{first.last}@tu-dortmund.de
http://www-ai.cs.tu-dormtund.de

Abstract. Due to IoT and Industry 4.0, more and more data is collected by sensor nodes, which send their data to a central data lake. This approach results in high data traffic and privacy risk, which we want to address in this paper. Therefore we use an existing Learning from Label Proportions (LLP) algorithm, to use the decentralized properties and extend this approach by applying Differential Privacy to the transferred data. This yields to reduced data transfer and increased privacy.

Keywords: Differential Privacy · Learning from Label Proportions · Distributed Learning · Spatio-Temporal · Traffic · IoT · Industry 4.0.

1 Introduction

Over the last few years decentralized data collection has become more and more popular. This trend is driven by IoT devices and its measured sensor values. In order to create added value for companies, this data is usually collected in a centralized manner, which results in high data transfer and data protection risks. Especially data protection risks are important to address by organizations, due to the introduction of GDPR in all EU countries in 2018 [5]. Organizations want to use this data in order to gain more information or predict future sensor states, e.g. "Will the traffic flow stay the same in the next 15-30 minutes?" and have to be compliant according to GDPR at the same time.

Therefore we extend the decentralized learning approach from [12, 13] by applying *Differential Privacy* to label proportions sent between the different decentralised IoT devices to result in a privacy preserving algorithm. Before we present our approach, we will first formally define the term *Differential Privacy*.

* This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038A) and by the German Research Foundation DFG under grant SFB 876 "Providing Information by Resource-Constrained Data Analysis" project B4 "Analysis and Communication for Dynamic Traffic Prognosis"

Differential Privacy Assumed, there is an algorithm $M : D \to R$ with domain D and range R. So D can be any set of input data and R the set of all possible outputs of M. Furthermore $D', D'' \in D$ have characteristic $||D' - D''||_1 \leq 1$, so that they do not differ in more then one element. This algorithm is ϵ-differentially private, if for all $S \subseteq R$ and for all D' and D'' the following formula applies [3]:

$$Pr[M(D') \in S] \leq e^{\epsilon} Pr[M(D'') \in S] \tag{1}$$

In more detail this definition formalizes, that the probability, that an algorithm M outputs the same results S by using different inputs D' and D'' differs in maximum by e^{ϵ}. Therefore by modifying the parameter ϵ, the degree of privacy, that will be applied, can be specified.

With $\epsilon = 0$, the probabilities of both outputs can be the same. This means that a single data point no longer has any influence on the output, making the algorithm completely privacy compliant. This is because the output does not allow any inference of a removed datum. However, you lose any information of the data, which makes the dataset useless.

Therefore $\epsilon = 0.1$ is usually chosen to increase the degree of privacy and still not reduce the information level of the data too much. The concrete implementation, how the algorithm becomes compliant with respect to equation, is described in section 3.

In this paper, we analyze the influence of adding noise to the label proportions. Therefore we will use the original *Learning from Label Proportions (LLP) algorithm* [12] and a simple centralized *k Nearest Neighbor (kNN)* [2, 10] classifier, to set benchmarks in aspects of accuracy and transferred data. We compare our results of the *modified LLP algorithm (p-LLP)*, that gains privacy aspects, with those benchmark results.

Firstly, we will give a brief introduction into other approaches, that can preserve privacy in machine learning (see section 2). Afterwards, we will describe the *LLP* algorithm and our modifications (see section 3), as well as analyze the performance on a city traffic dataset (see section 4). Finally, we will give a short conclusion.

2 Related Work

As already mentioned, data protection has become more relevant in the application of data-driven learning methods due to the GDPR. It has to be considered that decentralized data collection is the basis of the approach described here, as it allows to prevent the extraction of user data. There are various methods for increasing privacy preservation in case of decentralized data collection, such as data aggregation, data masking, or encryption. The current state of research for these methods is outlined in the following.

The basic idea of **data aggregation** is to hide individual data in the data of many. The data is collected over a longer period of time or from different data sources and aggregated using functions, which are usually additive. The resulting single characteristic per class contains thereby only one comparison

value, which can be compared with the other aggregated class values. Current methods that rely on aggregation for privacy protection include termSlice-Mix-AggRegaTe (SMART) [8], an improved version of this [16], the PriSense algorithm [11], or even the approach described in [6]. The latter does not aggregate the data, but the trained machine learning procedure. However, this method is only applicable if a decentrally trained model is used that can subsequently be updated globally. [8] and [11] partially send unique data to neighboring nodes, which reduces the security to be guaranteed, while [16] requires very high performance at the decentral nodes.

Data masking is another option for processing data that prevents private data from being read. For this purpose, the original data is enriched with certain values - so-called camouflage values -, or random values, so that the exact distribution, as well as the absolute data values, no longer correspond to the original. A simple approach for the aggregation functions minimum and maximum is described in [7]. The problem here is that Minimum and Maximum are sufficient just for very few use cases. The Cluster-based Private Data Aggregation (CPDA) approach [8] uses polynomial computations to aggregate within a cluster, but the reconstruction for the principal node of a cluster is very expensive. In contrast to this, the Federated Learning approach with Secure Aggregation described in [1] forwards updates of partial models to the server only in aggregate, using encrypted communication. Due to the partial encryption of the data and multiple data transfer it has high performance requirements as well, which often cannot be implemented in real life use cases. These high performance requirements are resulting from the partial encryption of the data and a multiple data transfer.

To increase data security through **encryption**, data is first encrypted, aggregated with other encrypted data, and finally decrypted again (secure aggregation [15]), whereas the type of encryption may vary. For example, the LVPDA approach [14] uses the Pallier Homomorphic Cryptosystem for encryption. Also [4] is based on this type of encryption and additionally uses a blockchain. As with all encryption-based approaches, both the amount of data to be transferred and the computing power requirements for the decentralized nodes are very high for these approaches.

In order to address the problems described above like insecurity or high performance requirements *LLP* approach was designed. This is achieved, by storing the measured data in a decentralized way. Each node stores its own collected data and only *Label Proportions*, are sent to neighboring nodes. This approach takes advantage of the fact that there is a correlation between distance of different measuring points and the predictive relevance. However, this aggregation does not guarantee that information about individuals is not disclosed. Therefore we have developed the following approach, which extends the *LLP* approach by differential privacy.

3 Algorithm

In general the *LLP* algorithm stays the same as proposed in [12]. Because it is

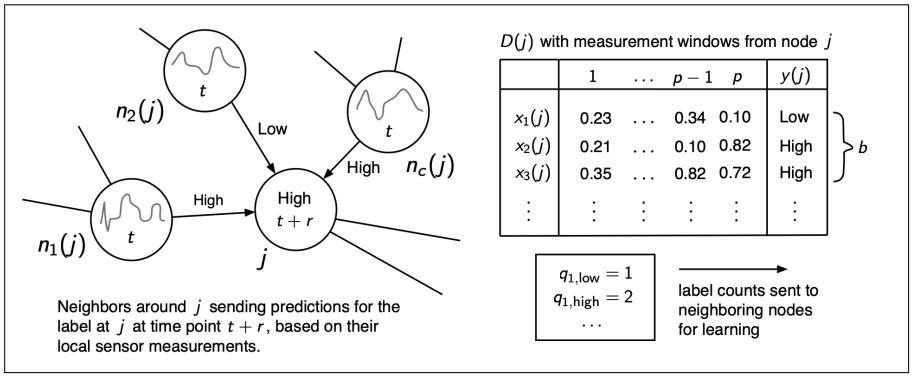


Fig. 1: Distributed Learning of Local Models, cited in verbatim from [12]

important to know the structure and flow of the algorithm, we will briefly discuss it by describing Figure 1. There are m wireless sensor nodes $(n_1, n_2, \ldots n_m)$, which store their measurements in $D(i) \forall i \in 1..m$. During explanation of the general algorithm and node setup, D denotes the measured data points ordered by window size and combined with the corresponding label as shown in Figure 1. Each row in $D(i)$ consists of $[t-w, t]$ measurements, where t denotes a timestamp and w is the *windows size* of the last w measurements. Each row is assigned a label, which is taken from a measured value from a future timestamp $t + r$. In the first place those measurements are split in batches $B_1, \ldots, B_h$ where $h = \lceil |D(i)|/b \rceil$ and b denotes the size of the batches, in which $D(i)$ will be divided. The *batch size* is also shown along the rows of the table in Figure 1. The *batches* are then used to calculate *label proportions* for each batch. The generated *label proportions* are sent to the closest c neighbors. Each node uses the received *label proportions* to train $c+1$ models $f_j(k)$, where $k \in 1, \ldots, c+1$ and j is the current node. The models $f_j(k)$ are learned by doing a k-means clustering on own measured node data $D(j)$. Initially each cluster, a random label is assigned. These cluster labels are exchanged by a more efficient local search with multi start strategy, introduced by [12]. In this phase, the label proportions of the neighboring node come in place for calculating the loss between the sent label proportions and the label proportions of the predicted labels, calculated by $f_j(k)$. Whenever the loss gets better, the exchanged cluster label assignment will be persisted and therefore result in a better model. The final prediction is done by doing a majority voting of the $c+1$ trained models.

This approach has the advantage, that we make use of more than only local measured datapoints, but keeping the bandwidth of transferred data low, because only *aggregated data* is sent between the nodes. However privacy cannot be guaranteed by this approach. Assuming, we have traffic flow measurement values, with labels $0, 1, 2, 3, 4$ and over a time frame of size b is only label 4 present. Then, from the label proportion, it can be inferred that everyone drove that fast during the period.

We solve this issue, by applying the *Differential Privacy* definition (see section 1) to the label proportions. Firstly, we have to calculate the l_1-sensitivity function to know, how much influence a single datapoint can make to the output of a function $f : D \to R$:

$$\triangle f = \max_{\substack{D',D'' \in D, \\ ||D'-D''||_1=1}} ||f(D') - f(D'')||_1 \tag{2}$$

For this scenario, D is the current batch B_i and R is the resulting *Label Count*. Considering that we have a simple counting query, a single datapoint can have a maximum influence of 1 (see [3] example 3.1). Finally we can use the *Laplace Distribution* to generate noise, which can be added to the *Label Counts*, to be privacy compliant under ϵ-Differential Privacy [3]:

$$lap(x, \sigma, \mu) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}} \tag{3}$$

$$lap(x, \frac{\triangle f}{\epsilon}, 0) = \frac{\epsilon}{2 \triangle f} e^{-\frac{|x-0|\epsilon}{\triangle f}} \tag{4}$$

In the formula above the *position* parameter μ is set to 0 and the *scale* parameter is set to $\frac{\triangle f}{\epsilon}$. This parameters have to be set like this, to be compliant with the *Differential Privacy* definition (proof can be found in [3] theorem 3.6).

The modified algorithm for calculating label counts can be seen in below. As mentioned before, the batches B_i are already generated and possible labels Y are also known. The output $Q(j)$ contains differentially private label proportions of all batches.

Require: $B_1, \ldots, B_h$, Y
Ensure: $Q(j)$
1: $Q(j) \leftarrow matrix(h, |Y|)$
2: **for** i **in** $1..h$ **do**
3: **for** j **in** $1..|Y|$ **do**
4: $Q(j)_{i,j} \leftarrow sum(B_i == Y_j)$
5: **end for**
6: // adding noise to label counts
7: $m \leftarrow sum(Q(j)_i)$
8: **for** j **in** $1..|Y|$ **do**
9: $Q(j)_{i,j} \leftarrow Q(j)_{i,j} + lap(e = 0, s = 1/\epsilon)$
10: clip $Q(j)_{i,j}$ to bounds $[0.001, m]$
11: normalize $Q(j)_i$
12: **end for**
13: **end for**

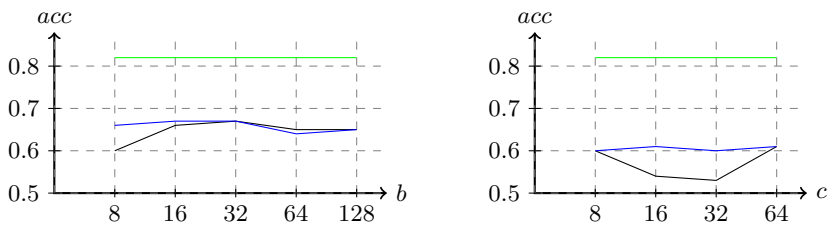
Initially $Q(j)$ is created with dimensions count batches (h) and count possible labels ($|Y|$). Afterwards the label proportions are calculated iterative for each batch as follows. Firstly, the label counts (see lines 3-5) and the total sum (see line

7) are calculated. Then the Laplace noise, which is calculated by the sensitivity and ϵ is applied. Afterwards the new value is clipped to the maximum bounds, to prevent to large, or negative values. Finally, the label counts with noise are normalized. The resulting proportion is stored in $Q(j)$.

4 Experimental Evaluation

For evaluation the traffic flow data at junctions in the city of Dublin, recorded by the Sydney Co-ordinated Adaptive Traffic System (SCATS) [9] is used. The dataset contains average traffic flow values for every 15 minutes in January 2013. To compute the needed sensor data $D(j)$ for the *LLP* algorithm, a sliding window with $w = 5$ is used. Corresponding labels $y(j)$ are the discretized traffic flow values at horizon $r = 1$. The discrete ranges are 0-5, 5-30, 30-60, 60-150, 150-260. The following test is based on this data of 4 sensor nodes and their 3 closest neighbors based on euclidean distance of geo-coordinates. All models are trained with 10-fold cross validation.

kNN with $k = 16$ is used, to give a reference, what accuracy results a centralised supervised learning approach can achieve. *LLP* is evaluated by varying the batch size $b \in \{8, 16, 32, 64, 128\}$, as well as varying the cluster count $k \in \{8, 16, 32, 64\}$ of integrated *k-means* algorithm. While modifying b the cluster count is fixed to $k = 16$, whereas when cluster count is modified, $b = 50$ is set. Both the cluster variations and the batch variations are performed with (*p-LLP*) and without (*LLP*) privacy applied. When privacy is applied $\epsilon = 0.1$ is set.



(a) batch size variations ($k = 16$) (b) cluster size variations ($b = 50$)

Fig. 2: Average accuracy results by applying different batch sizes (a) or different cluster sizes (b). Blue plot are results of *LLP*, the black ones are of *p-LLP* ($\epsilon = 0.1$). The green plot is *kNN* with $k = 16$ as benchmark

As shown in Figure 2 the average accuracy of *LLP* is in the range between 65% and 67%. The best results are achieved with a batch size of $b = 16$. By comparing the accuracy of *p-LLP* one can see, that results mostly differ when $b = 8$ is set. For batch sizes over 32, the accuracy varies by maximum of 0.5%,

which can happen due to tolerances. The results are understandable, since with small batch size the influence of a data point is proportionally large and therefore much noise is calculated on the label proportion, which yields in less information.

However the results in Figure 2 (b) are not so understandable. The cluster size is used to train the k-means algorithm, that is trained with measurement data from the own node. Therefore applying *Differential Privacy* to the variation of cluster sizes should not affect the average accuracy to drop by 6%.

As last evaluation step, we chose the best batch size $b = 32$ and best cluster size $k = 16$ from the previous evaluation, to analyze the influence of ϵ on *p-LLP*. As shown in Figure 3 the accuracy of 67% of *LLP* cannot be reached by *p-LLP*. However, it can be seen that with a very small epsilon ($\epsilon = 0.01$ or

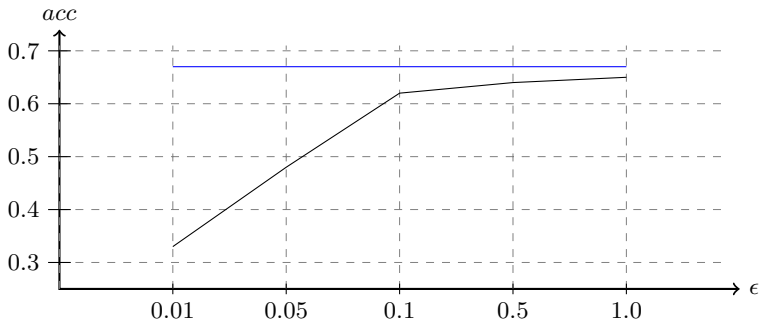


Fig. 3: Accuracy of *LLP* with $b = 32$ and $k = 16$ (blue). Accuracy of *p-LLP* with $b = 32$, $k = 16$ and varying ϵ (black)

$\epsilon = 0.05$) the dataset becomes unusable, which is evident from the accuracy of 32%, respectively 48%. From an $\epsilon = 0.1$ the prediction accuracy of 63% more and more reaches the accuracy of *LLP*. At the same time, sufficient privacy is still guaranteed with $\epsilon = 0.1$.

5 Conclusion

In this paper we extended the *LLP* algorithm by applying *Differential Privacy* to the label proportions, that are sent between nodes. Moreover we showed in the evaluation, that it is possible to achieve nearly the same accuracy as *LLP* when setting $\epsilon = 0.1$ for *p-LLP*. Therefore the *p-LLP* algorithm can be seen as superior, due to a more privacy preserving mechanism, which is important regarding GDPR. Next *p-LLP* has to be evaluated based on a larger node network, as well as more neighboring nodes. Transferability to other subject areas must also be examined.

References

1. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp. 1175–1191. ACM (2017). https://doi.org/10.1145/3133956.3133982, https://doi.org/10.1145/3133956.3133982
2. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
3. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014). https://doi.org/10.1561/0400000042, https://doi.org/10.1561/0400000042
4. Fan, H., Liu, Y., Zeng, Z.: Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain. Sensors **20**(18), 5282 (2020). https://doi.org/10.3390/s20185282, https://doi.org/10.3390/s20185282
5. Goddard, M.: The eu general data protection regulation (gdpr): European regulation that has a global impact. International Journal of Market Research **59**(6), 703–705 (2017)
6. Grama, M., Musat, M., Muñoz-González, L., Passerat-Palmbach, J., Rueckert, D., Alansary, A.: Robust aggregation for adaptive privacy preserving federated learning in healthcare. CoRR **abs/2009.08294** (2020), https://arxiv.org/abs/2009.08294
7. Groat, M.M., He, W., Forrest, S.: KIPDA: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks. In: INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10-15 April 2011, Shanghai, China. pp. 2024–2032. IEEE (2011). https://doi.org/10.1109/INFCOM.2011.5935010, https://doi.org/10.1109/INFCOM.2011.5935010
8. He, W., Liu, X., Nguyen, H., Nahrstedt, K., Abdelzaher, T.F.: PDA: privacy-preserving data aggregation in wireless sensor networks. In: INFOCOM 2007. 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 6-12 May 2007, Anchorage, Alaska, USA. pp. 2045–2053. IEEE (2007). https://doi.org/10.1109/INFCOM.2007.237, https://doi.org/10.1109/INFCOM.2007.237
9. McCann, B.: A review of scats operation and deployment in dublin. In: Proceedings of the 19th JCT traffic signal symposium & exhibition (2014)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
11. Shi, J., Zhang, R., Liu, Y., Zhang, Y.: Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems. In: INFOCOM 2010. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 15-19 March 2010, San Diego, CA, USA. pp. 758–766. IEEE (2010). https://doi.org/10.1109/INFCOM.2010.5462147, https://doi.org/10.1109/INFCOM.2010.5462147

12. Stolpe, M., Liebig, T., Morik, K.: Communication-efficient learning of traffic flow in a network of wireless presence sensors. In: Proceedings of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases (PDCKDD 2015) (2015)
13. Stolpe, M., Morik, K.: Learning from label proportions by optimizing cluster model selection. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III. Lecture Notes in Computer Science, vol. 6913, pp. 349–364. Springer (2011). https://doi.org/10.1007/978-3-642-23808-6_23, https://doi.org/10.1007/978-3-642-23808-6_23
14. Zhang, J., Zhao, Y., Wu, J., Chen, B.: LVPDA: A lightweight and verifiable privacy-preserving data aggregation scheme for edge-enabled iot. IEEE Internet Things J. **7**(5), 4016–4027 (2020). https://doi.org/10.1109/JIOT.2020.2978286, https://doi.org/10.1109/JIOT.2020.2978286
15. Zhang, W.: Secure data aggregation. In: van Tilborg, H.C.A., Jajodia, S. (eds.) Encyclopedia of Cryptography and Security, 2nd Ed, pp. 1104–1105. Springer (2011). https://doi.org/10.1007/978-1-4419-5906-5_639, https://doi.org/10.1007/978-1-4419-5906-5_639
16. Zhang, X., Liu, X., Yu, J., Dang, N., Qi, X., Zhang, Q.: Energy-efficient privacy preserving data aggregation protocols based on slicing. In: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCom/SmartData 2019, Atlanta, GA, USA, July 14-17, 2019. pp. 546–551. IEEE (2019). https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00109, https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00109