Enhancing IoT Anomaly Detection Performance for Federated Learning

Brett Weinger[1], Alex Sim (advisor)[2], John Wu (advisor)[2], Jinoh Kim (advisor)[2]
[1]Stony Brook University, [2]Lawrence Berkeley National Laboratory
Brett.Weinger@stonybrook.edu, {asim, kwu, jinoh}@lbl.gov

INTRODUCTION

Due to the ever-increasing concern of privacy and data confidentiality, a centralized data repository is not always feasible to conduct machine learning. Federated learning (FL) is a new distributed learning paradigm that uses a client-server architecture to train a global model using a weighted average of local updates computed on user devices without sending any private data to the server.

When FL is used in highly decentralized settings, learning can take a large number of iterative rounds to converge to similar performance levels of centralized models. In particular, when data on user devices possess the *class imbalance* issue, as is often the case for IoT anomaly detection, deep neural networks may initially ignore the minority class to achieve a reasonably low loss. This can be problematic for practical use of FL due to bottlenecks in transmitting information between the server and client nodes, and so we investigate algorithms to reduce the number of rounds necessary to achieve high performance metrics.

METHODS

We use the TON_IoT datasets[1], a recent collection of sensor readings for various IoT devices under normal and attack scenarios, to perform anomaly detection under FL. After partitioning this data across a fixed number of nodes, random oversampling, the Synthetic Minority Oversampling Technique (SMOTE), and the Adaptive Synthetic (ADASYN) method are each used to *augment* client datasets. Random oversampling will replicate samples present in the client's original dataset, while SMOTE and ADASYN generate a new example between an anomaly and one of its *k*-nearest neighbors in the feature space. ADASYN will bias the probability of choosing an anomaly in favor of ones that are closer to normal samples to attempt to make the classification boundary more well-defined.

Trials are performed under both *homogeneous* partitioning, where each client node receives an equal proportion of the total available data, as well as *heterogeneous* partitioning, in which we sample from a Gaussian distribution to determine what proportion of the remaining data a given client will receive. The latter approach is more realistic for real FL settings, while the former assumes that each node will have the same contribution to the global model during aggregation. Figure 1 shows baseline F1 scores in the homogeneous setting for various amounts of nodes, in which it takes a considerable number of rounds for this metric to become nonzero (i.e. to correctly recognize anomalies).

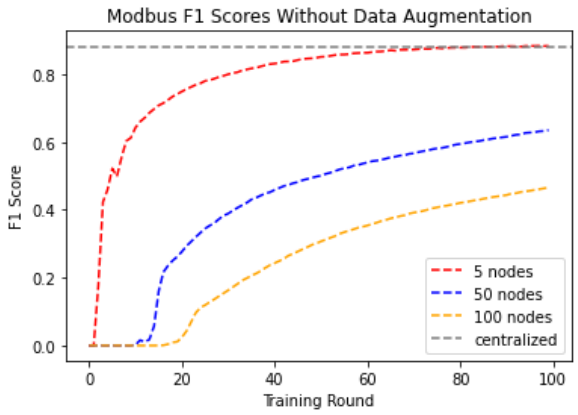

Figure 1: F1 scores for naive FL classifiers trained over 5, 50, and 100 nodes

RESULTS

Trials conducted for 5, 50, and 100 nodes in the homogeneous setting indicated that advantages of data augmentation are more pronounced when a greater number of clients participate in training rounds, each with a small proportion of the total dataset. Table 1 shows final accuracy, precision, recall, and F1 score metrics over a testing set after 100 rounds of training as well as the average computation time for each round. F1 score is a less biased metric than accuracy for this task as testing sets remain imbalanced, where a high value indicates that clients reliably and frequently detect anomalies. For 50 and 100 nodes, classifiers trained using random oversampling reach higher F1 scores quicker with only a small additional computational expense. This improvement is most apparent for 100 nodes, where random oversampling performs 19.98% better than the baseline with only a 6.95% increase in average round time.

[1]https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/

# Clients	Method	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Avg Round Time (s)
5 nodes	NONE	**94.74**	**86.99**	89.94	**88.39**	**13.58**
	RAND	94.65	86.49	90.24	88.33	23.35
	SMOTE	91.97	79.68	86.28	82.84	20.56
	ADASYN	87.91	67.07	**90.72**	77.12	24.77
50 nodes	NONE	85.98	76.47	54.36	63.55	**22.77**
	RAND	**87.29**	**77.97**	60.92	**68.40**	25.81
	SMOTE	76.30	48.07	**73.42**	58.10	24.80
	ADASYN	76.81	48.77	72.20	58.22	24.98
100 nodes	NONE	81.87	69.07	35.20	46.64	**34.68**
	RAND	**83.49**	**70.99**	46.17	**55.96**	37.09
	SMOTE	72.22	42.65	**66.88**	52.09	36.87
	ADASYN	73.34	44.20	65.46	52.77	36.63

Table 1: Metrics for various nodes and oversampling methods after 100 rounds of training

A benchmark of 70.00% is used to compare classifier performance for different augmentation methods. Figure 2 shows data collected over 100 nodes in training, where using random oversampling results in 211 rounds to reach this benchmark and the naïve classifier with no augmentation takes 280 rounds.

For the heterogeneous setting, 35 nodes were used such that available data would not be exhausted too quickly. The same Gaussian partitioning of the dataset was used for all oversampling methods. Figure 3 shows F1 scores for these trials, while Figure 4 shows F1 scores for an equal number of nodes in the homogeneous setting. While there is only a 0.98% difference between random oversampling and no augmentation after 100 rounds of training, the former classifier reaches the 70% threshold in 15 rounds as opposed to 34, which is more than a twofold decrease. Furthermore, it is notable that classification over heterogeneously partitioned data outperforms the homogeneous case for each augmentation method.

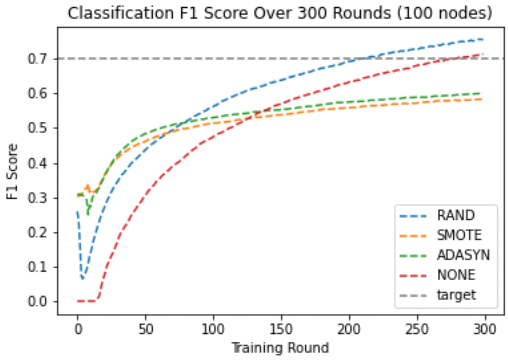

Figure 2: F1 scores for different oversampling methods relative to 70% target threshold, trained over 100 nodes

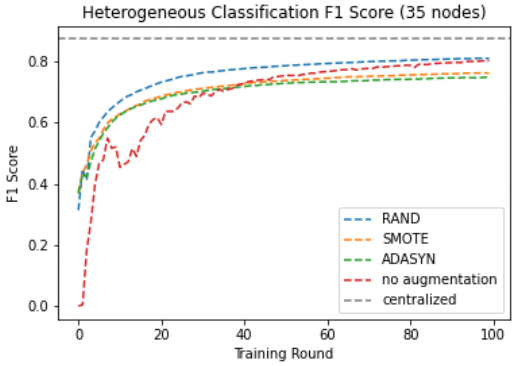

Figure 3: F1 scores under heterogeneous setting, trained over 35 nodes

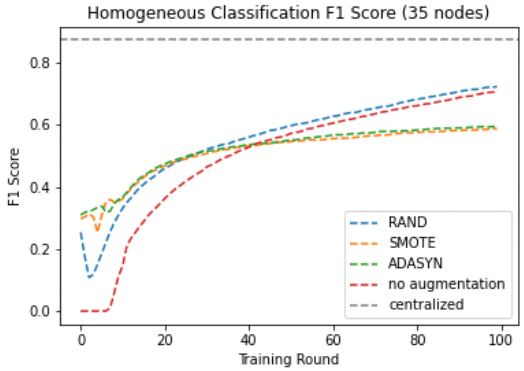

Figure 4: F1 scores under homogeneous setting, trained over 35 nodes

CONCLUSIONS

Data augmentation has been shown to yield a reduction in training rounds to attain an acceptable performance level of 24.6% in the homogeneous setting and 55.9% in the heterogeneous setting. Random oversampling outperforming other augmentation techniques suggests that anomalies are not highly clustered together, but rather sparsely distributed amongst the normal readings. This is likely to be common behavior in practice and motivates well-performing supervised classification to recognize such malicious behavior.

FL appears to be robust over heterogeneously partitioned data due to its weighted averaging scheme emphasizing updates from clients with more examples to use in training. So long as these examples follow the same distribution as data belonging to other clients, it can be advantageous for some local models to be assigned a higher weight than others.

ACKNOWLEDGEMENT

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and also used resources of the National Energy Research Scientific Computing Center (NERSC).

REFERENCES

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, pages 1273–1282, 2017.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, H Brendan McMahan, et al. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046, 2019.

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357, 2002.

[4] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. IEEE international joint conference on neural networks, 1322–1328. IEEE, 2008.