FedClamp: An Algorithm for Identification of Anomalous Client in Federated Learning

Habib Ullah Manzoor[1], Muhammed Shahzeb Khan[2], Ahsan Raza Khan[1], Fahad Ayaz[1], David Flynn[1], Muhammad Ali Imran[1], and Ahmed Zoha[1,*]

[1]*James Watt School of Engineering, University of Glasgow, United Kingdom*

[2]*School of Business Administration, Muhammad Ali Jinnah University, Pakistan*

Email: {h.manzoor.1, a.khan.9}@research.gla.ac.uk, shahzeb.khan@jinnah.edu.pk,

{Ahmed.Zoha, David.Flynn, Muhammad.Imran}@glasgow.ac.uk,

*Corresponding Author: Ahmed.Zoha@glasgow.ac.uk

***Abstract*—With the ever-increasing internet of things (IoT) and the rise of edge computing, federated learning (FL) is considered a promising solution for privacy and latency-aware applications. However, the data is highly distributed among several clients, making it challenging to monitor node anomalies caused by malfunctioning devices or any other unforeseen reasons. In this paper, we propose FedClamp, an anomaly detection algorithm based on the hidden Markov model (HMM) in the FL environment. FedClamp identifies the anomalous node and isolates them before aggregation to improve the performance of the global model. FedClamp was tested in a short-term energy forecasting problem using artificial neural networks when the FL environment had five clients. The algorithm uses mean absolute percentage error (MAPE) generated from local models and clusters them in normal and faulted nodes using HMM. The anomalous nodes identified using this algorithm are isolated before aggregation and achieve global model convergence with few communication rounds.**

***Index Terms*—ANN, Short term load forecasting, Federated learning, Anomalies, Energy forecasting**

I. Introduction

The value of data science in engineering is becoming more and more apparent as storage and processing capability are increasing day by day. Artificial intelligence, machine learning (ML), smart manufacturing, and deep learning in industrial engineering have all witnessed tremendous growth in recent years [1]. However, there are multiple challenges in developing future ML models. The data used in model training is highly privacy sensitive and protected under the Data Protection Regulation (GDPR) legislations [2]. Furthermore, the end users are more privacy aware; therefore, they are reluctant to share data. Data collection and storage are also costly and time-consuming. The problem with traditional ML is how the model is trained. The server for traditional ML typically manages data storage and model training. There seem to be typically two approaches to utilise these trained ML models [3]. We may either transmit the ML models to any device interacting with the environment or design a pipeline for the data to travel via the server [4]. Regretfully, neither of these methods is the best because their models are incapable of making quick adjustments [5]. To overcome these challenges, a new learning paradigm emerged termed federated learning (FL). In this learning approach, model training is done on an edge device under the supervision of a centralised entity without data sharing.

With the development of cheap and powerful electronic devices, next-generation sensors are capable of edge computing [6]. Combined with the growing concern of data privacy has given more importance to FL. The advantage of FL is distributed and continual learning, which can be achieved using multiple edge devices [7].

FL has been used in many areas, including mobile apps, IoT, transportation, and defence. FL is currently being used for emoji perdition in mobile phones [8], in smart devices to predict human trajectory [9]; in healthcare applications to proactively identify health anomalies [10] and predict hospital stay time [11]. Moreover, it also has applications in domain of energy network, smart homes and financial modeling [12]–[15]. Despite its ongoing development and use in a wide domain of applications, FL still faces many challenges. A few of the challenges are:

1) **High communication cost:** One of the bottlenecks of FL is the high communication cost associated with its use in a real-world application. [16].
2) **High number of communication rounds:** High number of communication rounds combined with huge ML models leads to increased latency.
3) **Data heterogeneity:** Since multiple devices are involved in FL, they might not follow identical independent distribution (IID) [17], which can create problems in model aggregation.
4) **Asynchronous aggregation mechanism:** The loss of internet connection on edge device can cause a problem in model aggregation [18].
5) **Anomalous edge device/client:** If one of the devices is suffering from an anomaly, it may mislead the server [19], and also his behaviour might impact other devices.

In FL, the ML model depends on all the devices, and an anomalous client can affect the devices involved in a network of devices by drifting the ML model. The anomaly can be due to malfunctioning of the edge device or any other reason. Identifying anomalous clients in FL can help us

protect edge devices and prevent any upcoming disaster from malfunctioning the device. This paper presents FedClamp, an algorithm for detecting and isolating anomalous nodes in FL before global model aggregation. The algorithm capitalizes mean absolute percentage error (MAPE) as a metric to flag node anomalies and Hidden Markov model (HMM) clustering to detect anomalous clients. We used the FedClamp to identify anomalous devices on the hourly energy data set obtained at the substation level.

II. Overview of Federated Learning

FL was designed to use ML in a decentralized manner, where data does not need to be collected on a central server. FL solves the problems associated with data island, and privacy [2]. Google reported the first use of FL in 2016, where they predicted the text entered by the user while keeping the data on edge devices [20]. A five-step baseline architect of FL is presented in Fig. 1. Firstly server will send a generic ML model to all edge devices. In the second step, the sent ML model is updated using the local data available on each device. This updated model will be called the local model. In the third step, all the edge devices will send the weights and biases of the ML model to the server. In the fourth step, the server will use some aggregation mechanism to combine all the weights and biases and make a new ML model. The most common aggregation method is FedAVG, where the weights and biases are averaged to make a new model [21]. This model will be called the global model. The global model will be sent back to all the edge devices in the final step. These five steps will complete one communication round between the server and clients. The entire process will run for pre-decided communication rounds.

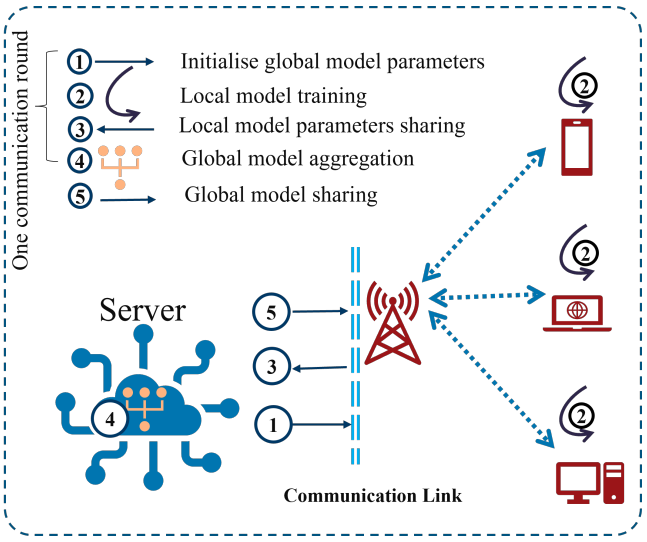

Fig. 1: An Overview of Federated Learning

III. FedClamp

The identification of anomalous clients in FL can help in the improvement of attaining a more accurate ML model, and it can also save computation power by reducing the number of communication rounds. The proposed algorithm will start by generating a generic ML model at the server. The model will be sent to the edge devices, where locally stored train data will be passed through the ML model to generate the local ML model. Here test data set will be used to test the accuracy of the ML model by calculating the performance evaluating parameter such as mean average percentage error (MAPE). The local weights, biases, and performance evaluating parameters will be sent to the server for aggregation, such as FedAVG. After completing all communication rounds, the server will find the euclidean distances of all the received MAPEs of one client with all other clients. If any of the euclidean distances is greater than the given threshold, it means there is an anomaly in clients. However, this will not identify the anomalous client. The anomalous client will be found by using HMM clustering on the sum of each client's euclidean distances. HMMs are probabilistic models introduced in the late 60s and are very helpful in clustering applications [22]. In this work, Gaussian HMM is used to cluster the normal and anomalous nodes, using the euclidean distance of MAPE at each client. In this approach, HMM takes the MAPE distance as a function of probability distribution function and transform it to given number of hidden states. In the given problem, the hidden state with lowest number of elements will represent node level anomalies.

The entire algorithm is summarised as:

Algorithm 1 FedClamp: Anomaly detection algorithm in Federated Averaging

Required: T = Communication Round, K = Client Count with index k, σ_k = MAPE loss on each client, B local batch size, E local client epochs, η = learning rate,

Server Execution:
Initialise the global weights w_0
for Each round $t = 1, 2, ...T$ **do**
  for Client count $k = 1, ...K$ **do**
    $w_{t+1}^k \leftarrow$ClientUpdate(σ_k, k, w_t)
    $\prod_t^k \leftarrow$Compute MAPE Distace (d) of each client
    if $MAPE(d) < threshold$ **then**
      Anomaly Flag = True
      HMM to identify anomalous node
    end if
  end for
  $w_t \leftarrow \sum_{c=0}^{K} \frac{n_c}{n} w_{ct}$
end for
$w_{t+1} \leftarrow w_t$

IV. Experiment and Results

We have evaluated the efficiency of our proposed algorithm on a real-world energy data set obtained from PJM Interconnection LLC [23], to identify the anomalous client in FL. Each column in the data set describes hourly energy consumption in a particular region. In this experiment, only one region is used.

Here five clients were created, and an artificial neural network (ANN) having three layers was implemented. Each

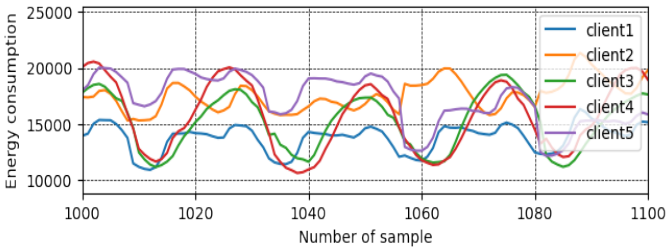

Fig. 2: Sample of data set.

client had 6288 samples. An original data sample is presented in Fig. 2. This data set is used for short-term load forecasting with the help of five features. The used features involve last hour data, last day same hour data, previous week same hour data, 24-hour average data, and last week average data. In ANN, Layer one had 100 neurons, layer two had 50, and the third layer had one neuron. All the layers had Relu activation function. The purpose of this ANN model is to predict energy consumption. The FL was carried out for 30 communication rounds with 20 epochs in each communication round. The results are presented in Fig. 3, where MAPEs of all clients are plotted with respective communication rounds. It can be seen that the graph converges around about eight communication rounds. The euclidean distances of all MAPEs are tabulated in table I. it can be seen that the highest euclidean distance Was 27.5 between client five and client two. Here, euclidean distance of 40, which is almost 1.5 times higher than the highest euclidean distance, can be assumed as the threshold for anomaly detection. Moreover, the highest sum of euclidean distances was 83.1 for client 2.

A. *One anomalous client*

To create an anomalous client, a tiny random seed was added to the actual weights of client one's first layer of the ML model. The addition of a random seed can be noticed in Fig. 4. It can be noticed that the graph converges around 17 communication rounds. The euclidean distances of all MAPEs are shown in Table II. The highest euclidean distance of 50.95 was obtained from clients one and four, which is greater than the decided threshold. Moreover, the highest sum of all euclidean distances was 168.15 from client one. HMM clustering was used to identify the anomalous client, as shown in Fig 5; here, green is used to identify the anomalous client.

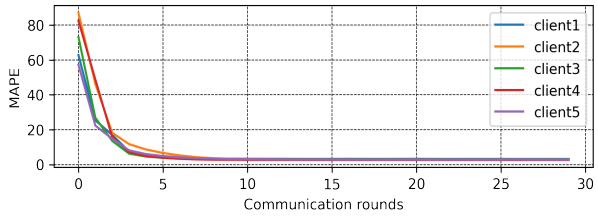

Fig. 3: MAPE of all clients without anomalies.

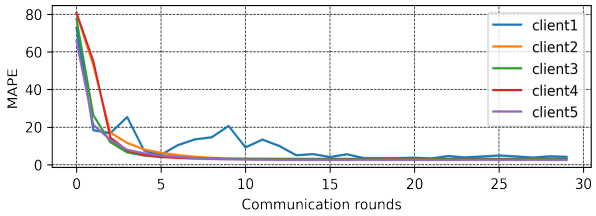

Fig. 4: MAPE of all clients with an anomalous client.

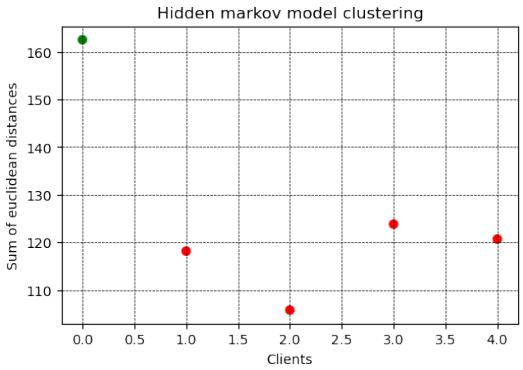

Fig. 5: HMM clustering with one anomalous client.

TABLE I: Euclidean distance of MAPE of each client

MAPE	Client1	Client2	Client3	Client4	Client5	Sum
Client1	0	25.8	8.4	23.2	3.3	60.7
Client2	25.8	0	21.1	8.7	27.5	83.1
Client3	8.4	21.1	0	17.8	10.1	57.4
Client4	23.24	8.7	17.8	0	25.3	75.0
Client5	3.3	27.5	10.1	25.3	0	66.2

TABLE II: Euclidean distance of MAPE of each client with an anomalous client

MAPE	Client1	Client2	Client3	Client4	Client5	Sum
Client1	0	47.12	35.4	50.95	35.04	168.1
Client2	47.12	0	28.05	7.6	35.4	117.2
Client3	35.4	28.05	0	29.14	13.22	105.8
Client4	50.95	6.7	29.14	0	37.04	134.7
Client5	35.04	35.4	13.22	37.04	0	120.6

TABLE III: Euclidean distance of MAPE of each client with two anomalous clients

MAPE	Client1	Client2	Client3	Client4	Client5	Sum
Client1	0	69.3	63.1	67.1	68.3	267.8
Client2	69.3	0	76.3	77.8	73.1	296.5
Client3	63.1	76.3	0	8.4	21.2	169.1
Client4	67.1	77.81	8.4	0	22.8	176.1
Client5	68.3	73.05	21.2	22.8	0	185.3

B. *Two anomalous clients*

In the second phase of experimentation, a random seed was added to the actual weights of the first layers of client one and client 2. The results are presented in Fig. 6 and the sum of all the MAPEs are tabulated in Table III. Comparing results with 3, the addition of a random seed is very prominent. Fig. 7, represented HMM clustering; here, green colour is used for anomalous clients. It can be noticed that HMM clustering has successfully identified anomalous clients.

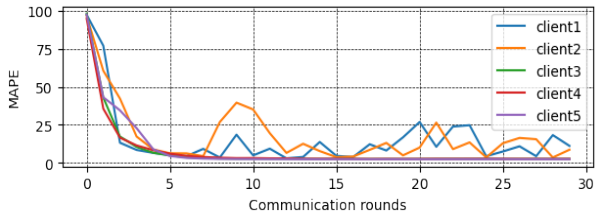

Fig. 6: MAPE of all clients with two anomalous clients.

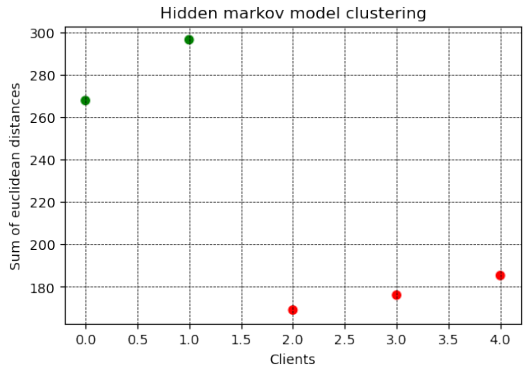

Fig. 7: HMM clustering with two anomalous client.

V. Conclusions

Recently, researchers have tilted towards the decentralised ML, known as FL, where ML models are sent to edge devices and data does not leave the edge device. FL has many advantages over centralised ML, such as reduced server cost, enhanced privacy, edge computing, etc. Since multiple edge devices are involved in the ML training process, it is essential to know the health of all the involved devices. This paper proposes FedClamp, an algorithm to identify the anomalous client in a distributed environment. The core idea of this algorithm is to find euclidean distances of MAPEs generated from local models and then use HMM clustering to identify anomalous clients. The algorithm was tested when the ML model was used to forecast energy consumption in the FL environment. We developed experiments in which we artificially injected anomalous nodes into the system and used FedClamp to identify these nodes.

References

[1] L. Li, Y. Wang, and K.-Y. Lin, "Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization," *Journal of Intelligent Manufacturing*, vol. 32, no. 2, pp. 545–558, 2021.

[2] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[3] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[4] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey of private distributed deep learning," *arXiv preprint arXiv:1812.03288*, 2018.

[5] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[7] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3241–3256, 2020.

[8] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv preprint arXiv:1906.04329*, 2019.

[9] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. IEEE, 2018, pp. 1103–1111.

[10] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[11] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal client behavior detection in federated learning," *arXiv preprint arXiv:1910.09933*, 2019.

[12] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Federated learning-based cognitive detection of jamming attack in flying ad-hoc network," *IEEE Access*, vol. 8, pp. 4338–4350, 2019.

[13] B. Hu, Y. Gao, L. Liu, and H. Ma, "Federated region-learning: An edge computing based framework for urban environment sensing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[14] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[15] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6283–6290.

[16] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[17] C. Godinho, J. Domingos, G. Cunha, A. T. Santos, R. M. Fernandes, D. Abreu, N. Gonçalves, H. Matthews, T. Isaacs, J. Duffen *et al.*, "A systematic review of the characteristics and validity of monitoring technologies to assess parkinson's disease," *Journal of neuroengineering and rehabilitation*, vol. 13, no. 1, pp. 1–10, 2016.

[18] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[19] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[21] Y. Zhou, Q. Ye, and J. Lv, "Communication-efficient federated learning with compensated overlap-fedavg," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 192–205, 2021.

[22] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[23] R. Mulla. Hourly energy consumption. [Online]. Available: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption