An Energy-saving Approach to Anomaly Detection Using Federated Learning

Pavan Raj Dasari
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
mdm77@mst.edu

Sahithi Thota
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
st8vf@mst.edu

Sai Ganesh Pamarti
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, United States
spbd3@mst.edu

***Abstract*— Anomaly detection is a critical task in various domains, such as healthcare, finance, and cybersecurity, where detecting rare events or abnormalities is of utmost importance and in machine learning that involves identifying abnormal events or patterns in data. Federated learning has emerged as a promising approach for anomaly detection, as it enables the training of a global model on data distributed across multiple devices without sharing the raw data. In this paper, we propose a novel approach for anomaly detection using federated learning, where each participant trains a local model on their data and shares the model updates with a central server. The central server aggregates the updates and sends a global model back to each participant. The local models then detect anomalies in their data based on the global model's predictions. The combination of anomaly detection and federated learning offers a promising solution for detecting rare and unusual events in real-world scenarios. This paper presents an overview of the current state-of-the-art in anomaly detection using federated learning, including different federated learning algorithms and architectures. We also discuss the challenges and open research directions in this area, such as privacy and security concerns, heterogeneous data distributions, and non-id data.**

Keywords—Anomaly, Anomaly detection, Federated learning, Isolation Forest, Fed averaging, Weight outlier detection, Energy Efficiency, Accuracy.

INTRODUCTION

Anomaly detection is the process of finding data points or occurrences that differ noticeably from most of the data. These anomalies could be signs of hostile activity, odd system or process behavior, or mistakes in data collecting. In many industries, such as finance, cybersecurity, and healthcare, anomaly detection is a crucial responsibility [8]. The detection of anomalies can be difficult since they can take many distinct shapes. Utilizing statistical techniques to model the data's distribution and spot data points that deviate from the norm is one form of anomaly identification. Another strategy is to use machine learning techniques to train a model on a collection of typical data and identify anomalies as data points that do not fit the model well [9].

In the realm of diabetic data mining, diabetes aberrant data detection has been a crucial study topic. Its goal is to uncover information in the vast database of diabetic patients that differs noticeably from other patients and raises enough doubt [1]. These odd points could hold valuable value information. Therefore, for doctors to make an accurate diagnosis, they must know how to extract abnormal information from diabetic patients effectively. Anomaly detection techniques currently available cover a wide range of applications, including common fraud, network intrusion detection, etc. [2] Utilizing the collective knowledge of distributed devices or nodes to detect anomalies while protecting privacy is the main goal of anomaly detection using federated learning. Using their respective data, the devices or nodes train their local models, which are subsequently combined into a global model by means of a secure aggregation protocol. Then, anomalies across the distributed data sources are found using the global model. The ability for companies to work together on anomaly detection without having to disclose their sensitive data with one another is one of the

major advantages of adopting federated learning for this purpose. This is crucial in sectors like healthcare and finance where strong data protection laws are in place.

The federated learning strategy is typically chosen in applications where data privacy is more important than model accuracy or where it is necessary for models to be trained independently on larger datasets. Google Gboard, healthcare, mobile computing, and credit card fraud detection among the current applications where federated learning has been employed. The key concerns in the healthcare sector are data security and privacy. Because healthcare data includes sensitive and valuable patient details such patient name, ailment and diagnosis, and other health-related characteristics, it is the most heavily targeted domain in terms of federated learning technique [5]. The machine learning model is stored on the server. It manages the distribution of the model across various clients and its training on locally stored datasets in the clients. Each client transmits its trained model to the server at each iteration. From the locally trained models, the server creates an aggregated model. The client devices are once more distributed with this aggregated model. The privacy of the user's data is ensured by the fact that the server does not have direct access to it. Every iteration of this procedure involves the construction of a new model using features provided by the user. After the requisite number of iterations, a final federated machine learning model is produced. While differential privacy is not guaranteed in a centralized architecture, it is in a federated approach where the data is guaranteed to be private because it is contained within the client devices [5].

Federated averaging, which is based on averaging local stochastic gradient descent updates for the primal issue, is the most often used technique for federated learning [14]. The federated averaging algorithm makes sure that each client device only communicates with the central server the updated model weights and never the raw data. This method safeguards the confidentiality of the local data and enables the effective training of models on sizable data sets dispersed over numerous devices or clients. The program also employs strategies like differential privacy to further safeguard the privacy of the local data [7].

A well-liked approach for finding anomalies or outliers in data is isolation forest. Data points that differ significantly from the rest of the data points are referred to as anomalies. The fundamental premise of the Isolation Forest method is that anomalies are usually rare and distinct, which enables them to be isolated from most of the data points by a limited number of straightforward selections. Isolation Forest is built on the notion of using binary trees to randomly divide the data into subsets. A random attribute and a random value for that attribute are chosen, and the data is divided according to those values to generate each subset. Up until each data point is isolated in its own partition, the process is repeated recursively. The program then calculates the average path length necessary to isolate each data point and utilizes this figure to provide an anomaly score. This measure is based on the premise that since anomalous data points are significantly different from most of the data points, isolating them will involve fewer steps [6].

RELATED WORKS

Anomaly detection can be performed using various machine learning techniques which includes K-nearest neighbors, Support vector machines, DBSCAN [Density-Based Spatial Clustering for Application with Noise], Autoencoders, Bayesian networks. These traditional anomaly detection algorithms have some drawbacks which includes the privacy issues where the data should be processed in a single location where we may have to expose the sensitive data, limited data access and high computational power. These drawbacks can be limited using federated learning anomaly detection.

Federated learning is one of the trending research topics. With sustainable development, federated learning has a lot of potential to find use in industrial engineering and healthcare. [12] Although it is a relatively new field of study, anomaly detection using federated learning has drawn attention because of its potential to overcome privacy issues when handling sensitive data. Federated learning maintains the privacy of the local data by keeping it private and only sharing updates to the model between servers or devices. The application of federated learning for anomaly detection has been examined in several papers. Using federated learning, for instance, a recent study developed a distributed framework for anomaly detection that enables many businesses to build a global model for anomaly detection without sharing their data. This framework's accuracy and privacy preservation performance on a dataset for credit card fraud detection showed promise.

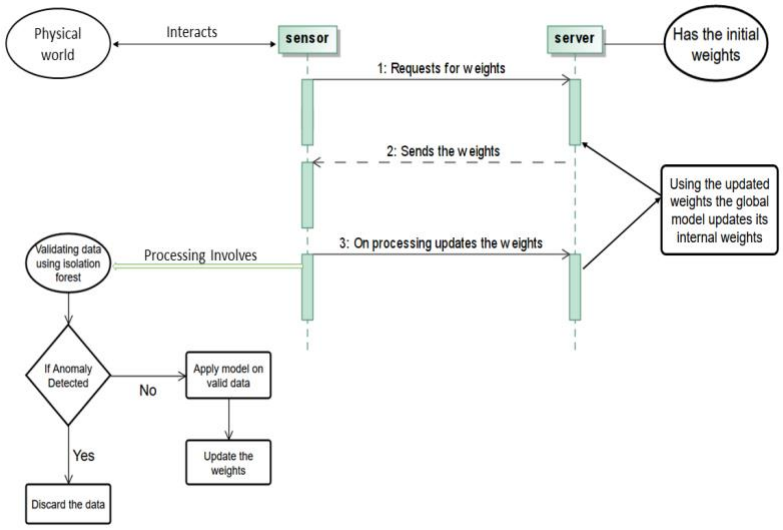
Another study suggested using federated learning to create a global model that can recognize irregularities in network data. This approach can be used to detect network intrusions. The accuracy of the proposed method was demonstrated to be superior to current state-of-the-art approaches while maintaining the confidentiality of the local data.[15]

A related study suggested using federated learning to discover abnormalities in power consumption patterns in smart grids, where different utility firms work together to do so. The suggested approach maintains data privacy while enabling each energy company to train a local model on its own data. The models are then integrated to create a global model, which is capable of identifying anomalies throughout the system.[16]

However, it is still difficult to train a global model using decentralized data because of the variability of the local data, transmission costs, and convergence problems. These issues must be resolved in order to increase the efficiency of federated learning for anomaly detection. The future of digital health may be solved by federated learning (FL), which also brings to light the issues and factors that need to be taken into account.[13]. So, we focused on the main issues which will eventually increase the energy efficiency.

WORKING MODEL

We consider a diabetes dataset which includes several attributes such as Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigree Function, Age, Outcome. The working model functions as follows: The server initially contains initial weights. Initially the sensor requests the server for initial weights, the server sends the weights to the sensor, and then the sensor starts processing. Processing is performed in the local model and the weights are updated. The updated weight from the local model is sent to the server which is the global model. The process is iterative until the system is accurate. Local model involves dataset validation using isolation forest. During validation, any anomalies will be discarded, otherwise the model will be applied to the data and the weights will be updated. And the global model uses fed Averaging for updating the model.



In this study, the diabetes dataset with 10k records was used to train a model using federated learning. The data was split into a train set (80%) and a test set (20%), and it was distributed among 10 clients for training. A global model was defined to run for 5 rounds. To ensure data quality, Isolation Forest was performed on each node to remove anomalies at the data level.

In isolation forest, initially we chose a sampling proportion from the original data set. The dataset includes various attributes such as glucose, Blood Pressure, Skin Thickness, Insulin, BMI, DiabetesPedigree Function, Age. To generate the binary decision tree in this isolation forest, we randomly select an attribute and its values. We repeat this process till the end of the sampling dataset. The main step is the calculation of the anomaly score at the data points. The anomaly score is calculated using the formula:

$$2^{E(h(x))/c(n)}$$

where h(x) indicates the number of edges in a tree at the data point
c(n) is the normalization constant for data of size n

If the anomaly score value is closer to 1 then we can say that the data point is anomalous. This means the value is extremely high or extremely low compared to the remaining values. If the values are closer to 0.5, the values are completely normal. Let's say the random attribute is glucose and all glucose values in the dataset range from 0 to 200. Using isolation forest, the anomaly score for values between 50 and 150 is close to 0.5, which means those are normal. Anomalies occur when the anomaly score is closer to 1, indicating anomalous values. Some anomalous values are 0,166,173,184,196. These are the values that cause sudden spikes/declines.

The new weights were calculated using an SVM model. Most of the traditional SVM algorithms provided by different libraries don't provide the ability to configure weights, so we have defined an SVM model which acts as a pretrained model which inputs initial weights sent from the server and fits the local data with those weights and produces new weights. Since the weights are already calculated, the model does not need to converge. Therefore, we have removed back propagation. The weights obtained from all the clients were fed-averaged to obtain a global weight. The global accuracy was then calculated, and the next round of training

was performed by repeating the isolation forest algorithm to remove anomalies, calculating weights, and obtaining the global weights. In the final round, an anomaly node was added, and the accuracy was calculated with and without the anomaly node. It is important to note that the success of this approach depends on the quality of the anomaly detection algorithm used and the distribution of the data among the clients. Additionally, privacy concerns must be considered to ensure that the local data is not compromised during the training process.

Algorithm 1 $FederatedAveraging$. *C* is the global batch size; *B* is the local batch size; the *K* clients are indexed by *k*; E is the number of local epochs; and η is the learning rate (adapted from [37]).

procedure SERVERUPDATE:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (subset *m* of clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}\,(k, w_t)$
    end for
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
  end for
end procedure
procedureCLIENTUPDATE(k, w)
  // Run on client *k*
  $B \leftarrow$ (split data *k* into batches of size *B*)
  for each local epoch *i* from 1 to *E* **do**
    for batch $b \in B$ **do**
      $w \leftarrow w - \eta \nabla F_k\,(w)$
    end for
  end for
  return *w* to server
end procedure

The algorithm we used is the Federated Averaging algorithm. Federated Averaging is a distributed optimization approach that is utilized in Federated Learning to train machine learning models on decentralized data. Without transferring or centralizing raw data on a server, the approach is made to enable effective model training. The initialization of a global model on a server, which is then distributed among numerous clients, is the first step in the Federated Averaging process. Then, without sharing their data with the server or other clients, each client trains the model locally using their own data. The clients submit the modified model parameters back to the server after the local training is finished. The server then creates a new global model by averaging the updated parameters it has received from each client. Until the global model converges to the appropriate degree of accuracy or the desired number of rounds are finished, this process is repeated iteratively. The technique ensures that the global model only receives the aggregated model parameters from the clients and never sees any raw data.

Energy efficiency is one of those parameters which we consider differentiating two methods. The amount of meaningful work or output generated for a specific amount of energy input is measured as energy efficiency. The ratio of output to input energy is a common way to express it, and a larger ratio denotes enhanced energy efficiency. Energy efficiency is calculated for both the traditional anomaly detection and anomaly detection using federated learning. The total energy efficiency of the traditional anomaly detection using SVM model is the summation of the energy efficiency for transmitting data from client to the server and the anomaly detection algorithm whereas the efficiency of the federated learning anomaly detection involves the energy efficiencies of transmissions from client to the server and vice versa and anomaly detection algorithm at local level which is isolation forest and fed averaging at global level. For this energy efficiency we had used some average parameters which includes the energy consumption, data transmission rate. Federated learning is performed for fixed number of rounds so for this some more parameters were considered which includes the number of rounds, number of clients, transmission power.

Based on some average values from internet, we define some of the parameters with their average values which includes transmission power as 10 milliwatts, distance between the devices as 100 meters, number of sensors as 10, number of rounds as 5, transmission power as 20 milliwatts. The total energy consumption of the network during data transfer operation is defined as 1000 Wh (watt-hours). The energy efficiency is combinedly calculated for data transmission, anomaly detection and model updations from client to server and server to the client. The size of the dataset is calculated in kilobytes (KB)A data transfer rate of 1 Mbps is defined, and the time it takes to transfer the data over the network is calculated in milliseconds (ms) using the formula: transfer_time = (data_size * 8) / data_transfer_rate * 1000.The total energy consumption of the network is converted from Wh to kWh (kilowatt-hours).The energy efficiency of the network is calculated in kWh per MB of data using the formula: energy_efficiency = total_energy_consumption_kWh / (data_size).

For fed averaging individually, energy consumed by one client per training round. This is done by first calculating the transmission time for data to travel from one device to another. This is based on data size, transmission power, and distance. The energy consumed by transmission is then calculated using transmission time, transmission power, and protocol efficiency. Using this energy consumption per client per round, the total energy consumed by all clients is calculated by multiplying by the number of clients and number of rounds.

For isolation forest individually first, we set the number of sensors and rounds. Isolation forest model with 100 decision trees and contamination='auto' parameter to detect anomalies. The code then measures the energy consumption of the system before the anomaly detection process using the psutil module to get the CPU usage percentage. The time measurement using the time module is also started. The model is then fitted to the synthetic data, and anomalies are detected using the predict() method. The system's energy consumption is measured again after the anomaly detection process. The total energy consumption is calculated in watt-hours (Wh) using the difference between the two energy measurements and the time elapsed during the process. The Isolation Forest model's energy efficiency is then calculated in kilowatt-hours (kWh) per data point. This is done by dividing the total energy consumption by the number of data points in the dataset (len(X)) and multiplying by the number of sensors and rounds.

For the traditional anomaly detection using SVM model the energy consumes for transmitting data from client to the server and for anomaly detection whereas for anomaly detection using federated the energy consumes for transferring the model weights from client to the server and server to the client, fed averaging and isolation forest anomaly detection. Though Federated learning uses the energy more effectively than regular anomaly detection.

RESULTS

The dataset we considered was the diabetes dataset. This includes several attributes such as glucose, blood pressure, skin thickness, insulin, body mass index [BMI], Diabetes Pedigree Function, and age. The dataset was divided into training and testing sets. We used 10 clients and five rounds of communication between the server and the clients. For every round we generate results. In round 1, let's examine the number of anomalies detected, the amount of data left after discarding data, and the accuracy of the model for each client. Here accuracy indicates how closely a calculated or forecasted value resembles the actual or anticipated value. We discard all anomalies detected at every client in every round. These data level anomalies are identified using an isolation forest. Based on the anomaly score we identify anomalies. If the anomaly score is nearer to 1, we discard the value considering those as anomalies, else if the anomaly score is nearer to 0.5 those data points are considered normal. At the end of every round the model is sent to the server where we calculate the global accuracy and the anomalous nodes. We look forward to eliminating all the anomalous nodes we manually inserted. At first, isolation forest did not detect the anomalies of this false node, as the anomaly score is close to 0.5, while the accuracy is only 49%. When the weights of this node are sent to the global server, the global server applies the weight outlier detection algorithm to the weights sent by the clients. It then performs the fed averaging, whereas the weight outlier detection algorithm detects the malicious nodes generating the false data. Here before applying the weight outlier detection algorithm the global accuracy is around 64%, but after applying the weight outlier detection the global accuracy is 96% which is effective. The presence of these anomalies results in incorrect predictions. Incorrect predictions include false results, even if they are true. The energy efficiency is calculated for both traditional anomaly detection and federated learning anomaly detection. For traditional anomaly detection the energy efficiency value is 102 kW/mb and for federated learning anomaly detection the energy efficiency is 124 kW/mb. The energy efficiency for the anomaly detection using federated learning is more than the traditional anomaly detection.

To summarize with, using the isolation forest and fed averaging we had identified the anomalies at data level and node level. Data level anomalies are identified using isolation forest based on the anomaly score values and node level anomalies are detected based on the global accuracy values. The data level anomalies are discarded at the local model itself rather than at the central server. The node level anomalies are discarded at server. We had also calculated the energy efficiencies for both the traditional anomaly detection and anomaly detection using federated learning.

CONCLUSION

Federated learning is a distributed learning technique that allows multiple parties to collaboratively train a machine learning model without sharing their data. Combining anomaly detection with federated learning can provide several benefits, such as improved data privacy, better model generalization, and reduced communication costs. By detecting anomalies locally on each participant's device and only sharing the results with a central server, federated anomaly detection can protect sensitive data while still enabling collaborative learning. This provides privacy and security since the data remains on the user's device and is not shared with the central server. This is particularly important for sensitive data like medical or financial records. Allows for the training of models on large-scale, diverse datasets without compromising data privacy, enables the use of machine learning algorithms that are more complex and accurate than traditional statistical methods, reduces the computational burden and communication costs associated with centralized training. Anomalies at data level and node level are focused. The energy efficiency is more for federated learning. However, further research is needed to address the challenges associated with this approach and to explore its potential applications in different domains.

References

[1] Ijaz M, Alfian G, Syafrudin M, et al. Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. Applied sciences 2018;8(8):1325.

[2] Zhang Y, Bingham C, Martinez-Garcia M, et al. Detection of Emerging Faults on Industrial Gas Turbines Using Extended Gaussian Mixture Models. International Journal of Rotating Machinery 2017;2017(1):1-9.

[3] Raed Abdel Sater and A. Ben Hamza. 2021. A Federated Learning Approach to Anomaly Detection in Smart Buildings. 1, 1 (June 2021), 24 pages.

[4] Dataset: https://github.com/plotly/datasets/blob/master/diabetes.csv

[5] Lincy, M., and A. Meena Kowshalya. “Early Detection of Type-2 Diabetes Using Federated Learning.” International Journal of Scientific Research in Science, Engineering and Technology (2020): 257–267. Web.

[6] F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.

[7] McMahan, B., Moore, E., Ramage, D., Hampson, S. & Arcas, B.A.y.. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data.

[8] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM Computing Surveys (CSUR) 41.3 (2009): 1-58.

[9] Chandola, Varun, and Vipin Kumar. "Anomaly detection for healthcare systems." Proceedings of the 2010 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2010.

[10] McMahan, H.B., Yu, F.X., Richtarik, P., Suresh, A.T. and Bacon, D., 2016, December. Federated learning: Strategies for improving communication efficiency. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain* (pp. 5-10).

[11] Sundaram, A., 1996. An introduction to intrusion detection. *Crossroads*, *2*(4), pp.3-7.

[12] Li Li , Yuxi Fan , Mike Tse , Kuo-Yi Lin. A review of applications in federated learning.

[13] Rieke, N., Hancox, J., Li, W. *et al.* The future of digital health with federated learning. *npj Digit. Med.* **3**, 119 (2020). https://doi.org/10.1038/s41746-020-00323-1

[14] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020, doi: 10.1109/MSP.2020.2975749.

[15] Rashid, Md Mamunur, Shahriar Usman Khan, Fariha Eusufzai, Md. Azharuddin Redwan, Saifur Rahman Sabuj, and Mahmoud Elsharief. 2023. "A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks" *Network* 3, no. 1: 158-179. https://doi.org/10.3390/network3010008

[16] Manzoor HU, Khan AR, Flynn D, Alam MM, Akram M, Imran MA, Zoha A. FedBranched: Leveraging Federated Learning for Anomaly-Aware Load Forecasting in Energy Networks. Sensors (Basel). 2023 Mar 29;23(7):3570. doi: 10.3390/s23073570. PMID: 37050631; PMCID: PMC10098660.

Contribution Breakdown

Content	Pavan Raj Dasari	Sahithi Thota	Sai Ganesh Pamarti
Project Research	40%	30%	30%
Code Implementation	35%	35%	30%
Data Collection	35%	35%	30%
Presentation	30%	35%	35%
Overall	34%	33%	33%