RESEARCH ARTICLE OPEN ACCESS

Oblique Federated Learning for IoT security in Disruption Tolerant Networks

Praveen Kumar[1], Suma Sekhar[2]
1(DoEEEVE, NIET, NIMS University, Rajasthan,Jaipur.
Email: praveen.kumar1@nimsuniversity.org)
2 (LBSITW, Kerala Technical University, Thiruvananthapuram
Email: sumasekhar@lbsitw.ac.in)

Abstract:

Delay Tolerant Networks are used in Wireless Sensor Networks based IoT Platforms. Though we adopt federated learning scenario for ensuring security, we can think of adding additional layer of security in which oblique platforms are considered. Byzantine-Resilient aggregation ensures that a few malicious client nodes cannot corrupt the global learning process.

Keywords **—Delay Tolerant Network (DTN), Federative Learning (FL), Oblique Federative Learning (OFL), Edge Computing, Data Aggregation, Mean Square Error(MSE), Store-and-Forward, Krum aggregation, Multi-Krum Aggregation, Byzantine client**

I. INTRODUCTION

The disruption tolerant Networks (DTN) are the class of Wireless Sensor networks where network nodes are not always in constant connectivity. DTN based connectivity is probabilistic in nature and we use bundled protocol for establishing connectivity. The focus of the study is related to Intrusion detection or abnormal behaviour in real time using Oblique Federated learning in such networks.

. Delay Tolerant Networking (DTN) and Oblique Federated Learning (OFL) are not inherently related, they can complement each other in environments where network connectivity is intermittent and device capabilities are heterogeneous. By combining the strengths of DTN in handling connectivity issues and the optimization strategies of OFL for heterogeneous devices, it's possible to create more robust and efficient distributed learning systems in challenging conditions

II. DELAY TOLERANT NETWORKS

DTN is a networking approach designed to operate effectively over extreme distances such as those encountered in IoT networks or in space communications or in environments with intermittent connectivity. Key features of DTN include (i)intermittent connectivity (ii) store-and-forward mechanism and (iii) tolerant to high delays.

Data is temporarily stored at intermediate nodes until a forwarding path is available. Optimised for environments where delays are long and unpredictable.

Decentralised Modal training is a highlight of a DTN network. By virtue of its inherent nature, training machine learning models across multiple devices without sharing raw data enhancing privacy and security.

III. OBLIQUE FEDERATED LEARNING

Oblique Federated Learning (OFL) is a variant of Federated Learning (FL) that focuses on optimizing learning processes when participating devices have heterogeneous computational and communication capabilities. Key features of OFL include

A. Comparison of ML, FL and OFL

A comparison of the traditional machine learning, federated learning and oblique learning methodologies are considered. Though the communication overhead is there OFL will be a boon in devising the learning strategy for real world DTN systems.

OFL model uses oblique splits. This increases expressiveness over federated learning. For doing this the communication overhead is not paid for.

TABLE I
COMPARISON OF ML,FL AND OFL

Feature	Device Learning Scenario		
	ML	FL	OFL
Device Location	Centralised	Distributed	Distributed
Privacy	Low	High	High
Model Type	Any	Any	Oblique Models (Eg.Trees)
Decision Boundaries	Depends on Model	Simple, Axis aligned	Oblique(Linear combinations)
Expressiveness	High	Medium	Higher than standard FL Trees
Complexity	Low-Medium	Medium	High
Communication O/H	Low	Medium	Medium-High

B. Simulation using Heterogenous datanodes

. The connectivity probability models with intermittent connectivity is considered. Each device trains a local model for a few epochs. The local model values over the rounds are plotted for **simple aggregation**. The local model value tries to catch up the *ideal model weight* which is set as **2.0** here. The accuracy metric taken here is the mean square error (MSE). The ideal value of MSE is zero.

The formula for calculating MSE is:

$$MSE = \frac{1}{n}\sum_{n=1}^{n}(y_{\iota} - \hat{y}_{\iota})^2$$

Where n is the number of data points.

y_i is the actual value.

$\hat{y}_{\iota}$ is the predicted value.

A decreasing loss indicates that the model is *learning effectively*.

Simple federative learning strategy uses simple averaging of local weights and updating global model. The oblique model uses median approach.

In the simulation without Federative learning (FL) each node trains its local model independently, and the average model value and accuracy are recorded.

In the simulation with FL models are trained locally, then aggregated to form a global model, which is evaluated.

The accuracies of the local and global models over training rounds are plotted for comparison. The Python code which provides a basic simulation and visualization is run it in a Python environment

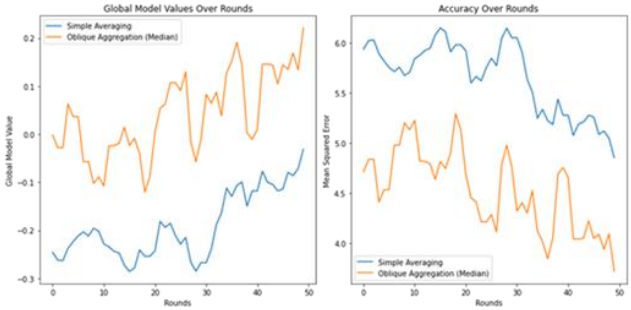

Fig. 1 Comparison FL and OFL Modal value and MSE

C. Byzantine Aggregation

In Oblique Federated Learning (OFL), the choice of distance metric is not just a mathematical detail. It directly determines how well we can separate honest against malicious client updates after transformation. The important distance vectors considered in machine learning parlance are

1) Euclidean Distance: This distance calculation is geometric distance assuming the feature vectors are independent. The real scenario is far away from this approximation

2) Manhattan Distance: It measures movement along axes only. Not suitable for oblique and non-aligned data distributions.

3) Mahalanobis distance: Measures distances considering variance and correlation.

$D_m = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}$ where is x is the datapoint.μ is the mean vector of distribution.

S^{-1} is the inverse of co-variance matrix

The Byzantine Aggregation doesn't utilises blind averaging. Aggregation utilses '*Mahalanobis distance'* for robust aggregation resisting attacks. It also demonstrates outlier rejection and thereby it reflects real federated learning security behaviour.

D. Comparison of Oblique Federated Learning Aggregation

The oblique Federated Learning aggregation techniques are compared here.

Krum is a robust aggregation rule introduced to defend federated learning against Byzantine (malicious) clients.

Fig. 2 shows an implementation of Krum algorithm in an IoT framework.

- the aggregation error for mean is more,
- the higher values shows influence of outliers or attacks
- Krum variants are lower and smooth, and it shows robust aggregation filtering out bad up dation.

TABLE 2
COMPARISON OF KRUM AND MULTI KRUM

Oblique Federated Learning Aggregation		
Feature	**Krum**	**Mult-Krum**
Update	Selects one client update	Select multiple reliable updates
Nature of updation	Ignores useful information from other honest clients	Reliable updates are taken and the score is averaged
Stability	Less	More
Noise	More	Less

Multi-Krum aggregation is an extension of the Krum aggregation algorithm used in Byzantine robust federated learning designed to be more stable and less noisy

The algorithm is like this.

- Compute pairwise distances
- Compute Krum scores for all clients
- Select top **m** clients
- Average their updates

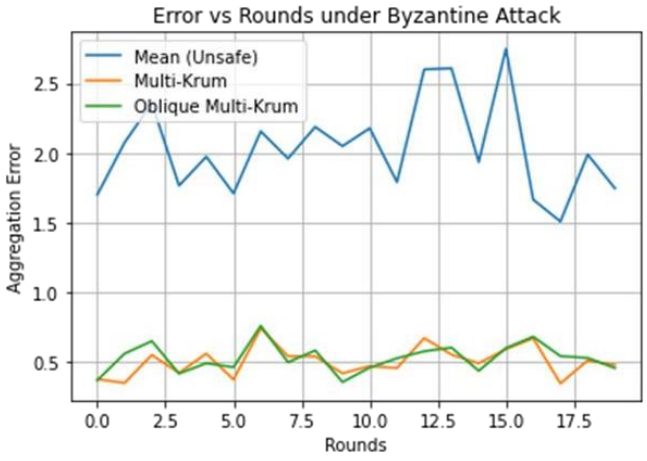

Fig. 2 Error Vs rounds under Byzantine attack

.

- The mean aggregation exhibits larger fluctuations and consistently higher values, indicating its vulnerability to adversarial or noisy client updates which significantly distorts the global model.
- In contrast Multi-Krum and Oblique-Krum maintain much lower more stable trajectories, demonstrating their robustness against such malicious influences. Multi-Krum achieves this by selecting only the most consistent updates based on Mahalanobis distance metric.

IV. METRICS COMPARED

A Aggregation Error analysis

The figure 3 represents the comparative performance of three aggregation techniques: Mean (Unsafe), Multi-Krum, and Oblique Multi-Krum under adversarial conditions.

. It is evident that the Mean (Unsafe) method produces a significantly higher value compared to the other two approaches. In contrast, both Multi-Krum and Oblique Multi-Krum yield substantially lower and closely comparable values.

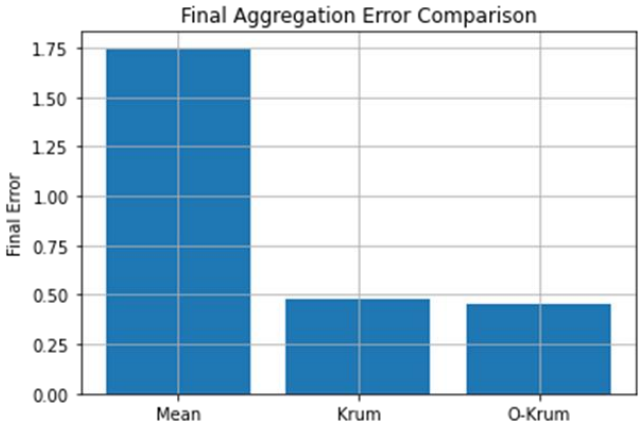

Fig. 3 Final Error Comparison

The higher value associated with Mean aggregation indicates its susceptibility to Byzantine or malicious client updates. Since this method computes a simple average, it fails to distinguish between reliable and adversarial contributions, resulting in distorted global updates

B Stability Comparison

. The bar chart in Fig 4 clearly shows that the **Mean (Unsafe)** method exhibits a significantly higher value compared to the other two methods. In contrast, both **Multi-Krum** and **Oblique Multi-Krum** demonstrate much lower and closely aligned values.

The elevated value for the Mean aggregation indicates that it is highly influenced by malicious or Byzantine client updates. Since simple averaging treats all client updates equally, even a small number of adversarial contributions can skew the global model significantly

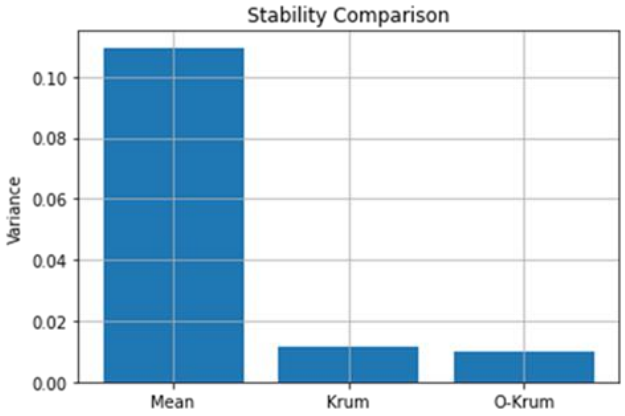

Fig. 4 Variance (Stability) Comparison

V. CONCLUSION

In edge computing scenarios with IoT devices, DTN can ensure that model updates are eventually delivered even if devices temporarily lose connectivity. In *remote or rural areas* with sporadic internet access, DTN can help OFL by ensuring data and model updates are stored and forwarded when possible, maintaining the integrity and progress of the learning process.

Integration Of DTN and OFL for better performance can be done by incorporating Mult-Krum algorithm and thereby the chances of loosing useful information from other honest clients are ruled out.

ACKNOWLEDGMENT

I acknowledge NIET, NIMS University, Jaipur for creating a wonderful academic atmosphere in the institution so that I can nurture the research in varied engineering topics including the security in IoT and Industrial IoT along with my dear final year Electronics and Communication Engineering students.

I acknowledge, Director NIET, NIMS University for his constant motivation and guidance for doing this research in IoT security domain.

REFERENCES

1. *Kevin Fall, Intel Research, Berkely "A delay Tolerant Network Architecture for Challenged Internets", SIGCOMM'03, August 25-29,2003, Karlsruhe, Germany*

2. *Kevin Fall, Intel Research Berkeley, Rabin Patra, University of California, Berkely Sushant Jain University of Washington, "Routing in a Delay Tolerant Network", Sigcomm'04, Aug-30-Sept3,2004 Portland, Oregon, USA*

3. *Tamer Abdelkader, Kshirasagar Naik, Amiya Nayik, Nishith Goel and Vineet Srivastava "A performance comparison of DTN routing protocols"*

4. *Stephen Farrel,Vinny Cahill ,'Chapter 3 Application Requirements for DTN' in the book 'Delay and Disruption-Tolerant Networking', Artech House, London*

5. *Stephen Farrel, Vinny Cahill ,'Chapter 4 Bundle Protocol' in the book 'Delay and Disruption-Tolerant Networking',Artech House, London*

6. *"Data Fusion and Processing in Wireless Multimedia Sensor Networks: An Analysis for Surveillance Applications." 2014 IEEE 22nd Signal Processing and Communications Applications Conference (SIU 2014).*

7. *Athanasios V. Vasilakos, Yan Zhang, and Thrasyvoulos Spyropoulos,"Delay Tolerant Networks"*

8. *Aloizio Pereira da Silva, Scott Burleigh, and Katia Obraczka."Delay and Disruption Tolerant Networks : Interplanetary and Earth-Bound - Architecture, Protocols, and Applications"*

9. Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong "Federated Learning"

10. *Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.*