Wrapper-Based Federated Feature Selection for IoT Environments

University of Kentucky

Presenter: Afsaneh Mahanipour

Faculty Mentor: Dr. Hana Khamfroush

Computer Science Department

Introduction

- Novel Internet of Things (IoT) applications have emerged as enabling technologies for the smart city initiative.
- IoT devices usually collect or produce huge multi-modal data, and these collected data need to be processed to create useful knowledge about the environment.
- Due to the inherent computation and storage limitations of IoT devices, the collected data is either processed on distributed edge servers or sent to a powerful and central cloud to be processed using novel machine learning methods.
- In these applications, each device has its own local data set, and the whole data set is scattered in a distributed nature. For example, in autonomous driving systems, there is a set of autonomous vehicles with a number of sensors, and each sensor collects its own local data observing its surrounding traffic.
- Transferring all these data to the edge servers needs a lot of communication cost. Moreover, they contain heterogeneous, noisy, and redundant data. Therefore, data pre-processing techniques such as feature selection methods can be used to decrease data size and improve communication cost.
- Feature selection methods select informative features among the original features in a feature set.

Fig. 1: Smart City

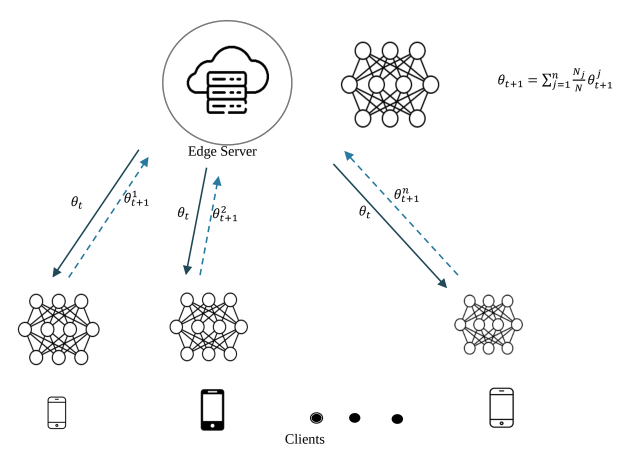


Fig. 2: Architecture of Federated Learning

Proposed Method

- A novel wrapper-based federated feature selection (FFS) algorithm, called BGSAFFS, is proposed, where IoT devices collaborate to select the most informative features without sharing their local data sets.
- The proposed FFS algorithm uses binary gravitational search algorithm (BGSA) in a federated and collaborative manner to select a small enough subset of informative attributes and provide an improved trade-off between communication cost and learning accuracy. Therefore, our goal is to train a model that selects best features.
- Inspired by original version of federated learning (FL) algorithm to train the global BGSA in an edge server, the local BGSA should be executed in clients and select best features from their datasets.
- Then global BGSA continues the process in an edge server to aggregate the selected features.
- Training of a global BGSA algorithm is performed in an iterative fashion. It communicates with local ones iteratively till the stopping criterion is reached.

Algorithm 1 Pseudo code of the edge server side

Input: M (number of clients) as the number of agents, Number of features, local "Best" solutions and their corresponding fitness values, and their index from all clients
Output: The "Best" solution (Best subset of features)

```
while the stopping criterion (or maximum iteration) is met do
    for all clients do
        execute Algorithm 2
        receive the best agent of a client and its corresponding fitness value and its index
    end for
    Global population = all local best agents of all clients
    Global fitness = corresponding fitness value of the best agents
    Updating Kbest, G, Best, Worst, and M
    Calculating the acceleration and velocity of global population
    Updating global agents' position
    send the updated best agent to its corresponding client
end while
```

Algorithm 2 Pseudo code of the client side

Input: The number of agents, Number of features, and the updated "Best" agent of the client from the global algorithm
Output: The local "Best" solution and its corresponding fitness value, and its index

```
Initialization: Generate an initial population of binary strings randomly
In iterations > 1, use saved population from the previous step and replace the updated "Best" agent
while the stopping criterion (maximum iteration) is met do
    Fitness evaluation of agents
    Updating Kbest, G, Best, Worst, and M
    Calculating the acceleration and velocity of the local population
    Updating agents' position
end while
save the position of local agents
```

Data and Results

- The proposed method is examined on MNIST, Fashion-MNIST (F-MNIST), and MAV data sets, and the results on these data sets have been reported.
- According to the literature, these three data sets reflect the characteristics of IoT data.
- The details of these data sets are provided in Table I.

TABLE I: Characteristics of the data sets

Data set name	Classes	Features	Instances
MNIST	10	784	70000
F-MNIST	10	784	70000
MAV	58	2166	2911

- Performance evaluation metrics are defined as classification accuracy (CA) and feature reduction (FR):

$$CA = \frac{C_N}{T_N} * 100$$

where C_N is the number of records in the test set that are correctly classified, and T_N is the total number of records in the test set.

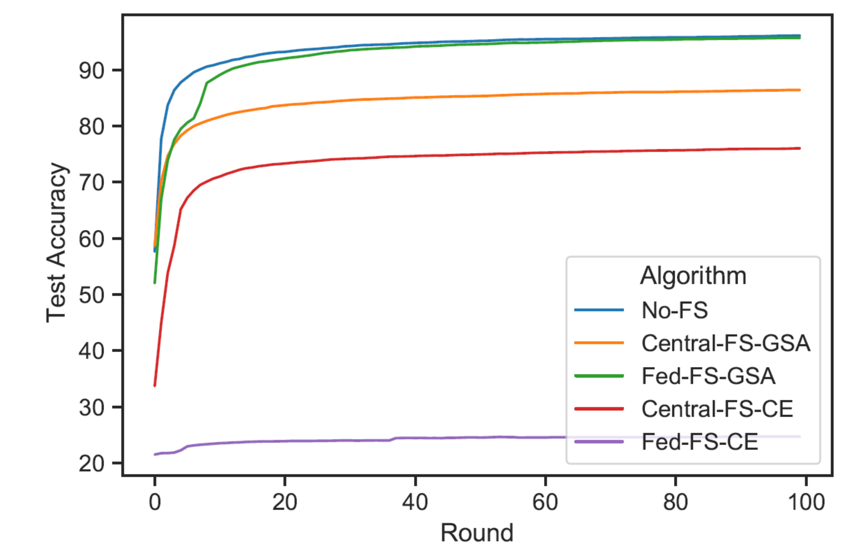
$$FR = \frac{TF - SF}{TF} * 100$$

where SF is the number of selected features, and TF is the total number of features in the original feature set.

TABLE II: Comparison between the proposed method and No-FS and method proposed in [2]

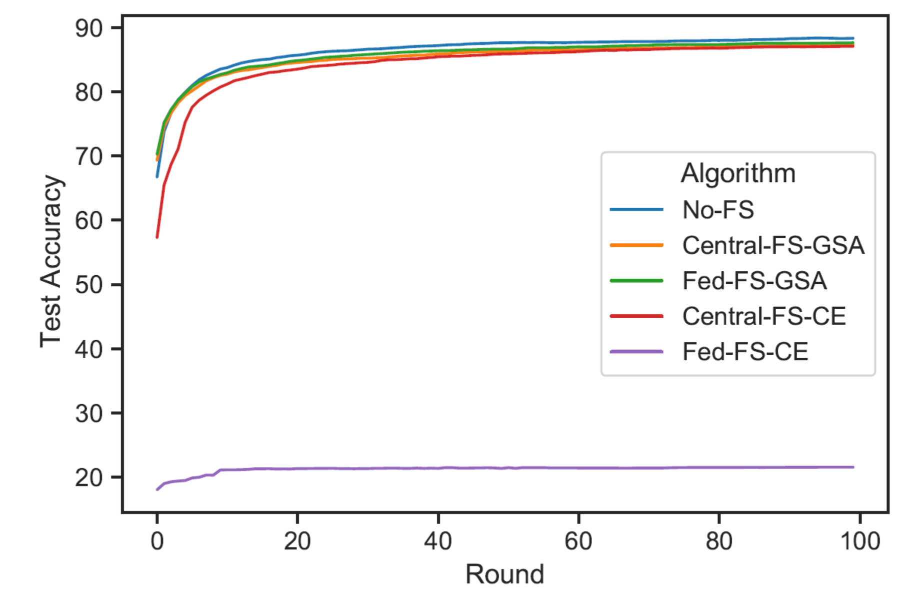
Data set	Method	Number of Features	FR (%)	CA (%)
MNIST	No-FS	784	-	96.10
	Fed-FS-BGBA	400	48.97	95.80
	C-FS-BGBA	385	50.89	86.43
	Fed-FS-CE [2]	9	98.85	24.70
	C-FS-CE [2]	308	60.71	76.03
F-MNIST	No-FS	784	-	88.33
	Fed-FS-BGBA	405	48.34	87.62
	C-FS-BGBA	382	51.27	87.30
	Fed-FS-CE [2]	2	99.74	21.53
	C-FS-CE [2]	289	63.13	87.08
MAV	No-FS	2166	-	68.26
	Fed-FS-BGBA	1068	50.69	67.71
	C-FS-BGBA	1112	48.66	67.03
	Fed-FS-CE [2]	19	99.12	33.51
	C-FS-CE [2]	665	69.29	61.67

- MNIST has 784 features. This figure implies that our proposed algorithm, Fed-FS-GSA, achieves same test accuracy as No-FS while reducing communication cost by using only 51% of features for MNIST dataset.



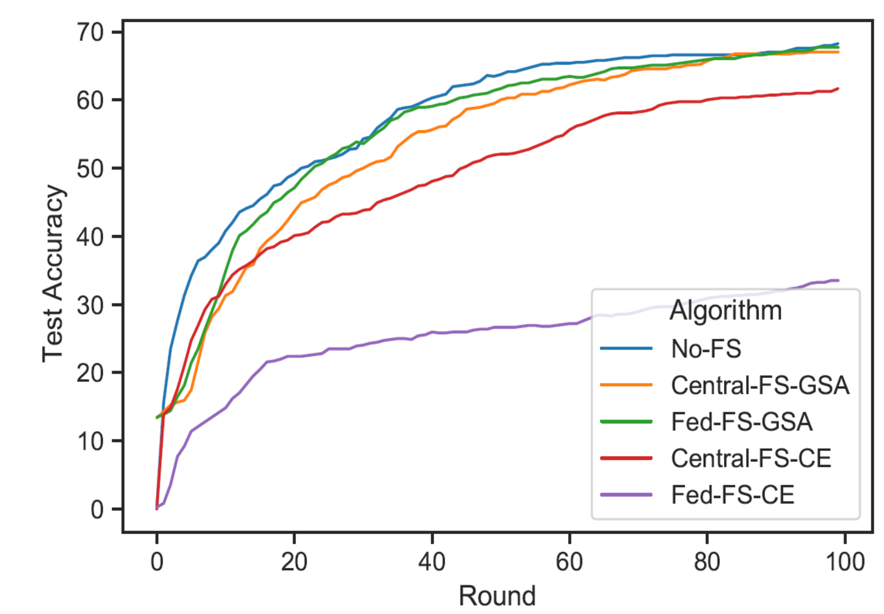
MNIST Dataset

- Fashion MNIST (F-MNIST) has same number of features as MNIST. Our results represents that our proposed algorithm, Fed-FS-GSA, reduces communication cost by removing 48% of features while achieves same test accuracy as No-FS with the error rate of less than 1%.



F-MNIST Dataset

- MAV dataset has 2166 features. Our proposed algorithm achieves same test accuracy with the error rate of less than 0.5% while improving the communication cost by removing about 51% of features.



MAV Dataset

Fig. 3: Accuracy of the proposed method for three different data sets

Conclusion and Future Work

- In this work, a wrapper based federated feature selection method with binary gravitational search algorithm is proposed for the first time.
- The experimental results on three benchmark data sets demonstrate that the proposed BGSAFFS beneficial for providing suitable and small enough feature subsets.
- For example, in MNIST data set, the FR of the proposed method is almost 49% and its accuracy is 95.80 which is almost equal to the accuracy of the classifier with no feature selection method.
- In the proposed method, number of selected features cannot be considered as a predefined parameter.
- Some applications may need a certain level of accuracy. Therefore, we intend to investigate using hybrid FS methods to prioritize features and then choose a specific number of features as a future work.

References

1. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019, pp. 1–7.
2. P. Cassar`a, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," arXiv preprint arXiv:2109.11323, 2021.
3. B. Venkatesh and J. Anuradha, "A hybrid feature selection approach for handling a high-dimensional data," in Innovations in computer science and engineering. Springer, 2019, pp. 365–373.
4. T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, 2020.
5. Ye, H. Li, A. Imakura, and T. Sakurai, "Distributed collaborative feature selection based on intermediate representation." in IJCAI, 2019, pp. 4142–4149.
6. Rashedi, H. Nezamabadi-Pour, and s. Saryazdi, "Bgsa: binary gravitational search algorithm," Natural Computing, vol. 9, no. 3, pp. 727–745, 2010.