Multimodal Multiple Federated Feature Construction Method for IoT Environments

University of Kentucky

Presenter: Afsaneh Mahanipour

Faculty Mentor: Dr. Hana Khamfroush

Computer Science Department

Introduction

- Smart devices and new Internet of Things (IoT) applications have been developed very fast during these years.
- A huge volume of data can be collected by these devices and fed into different machine learning (ML) models to extract knowledge about the environment.
- Because of real-time response requirements and privacy issues, for many IoT applications the data cannot be sent to a central cloud server and need to be processed either locally or on an edge server.
- These data are high dimensional big data that may contain some irrelevant, redundant, noisy, or heterogeneous ones.
- Federated feature construction as a data pre-processing method can be applied on local datasets of each device to reduce data size and consequently, improve the model performance and communication cost.
- Feature construction methods construct high-level features by combining informative features with suitable operators to extract hidden relationships among original features.

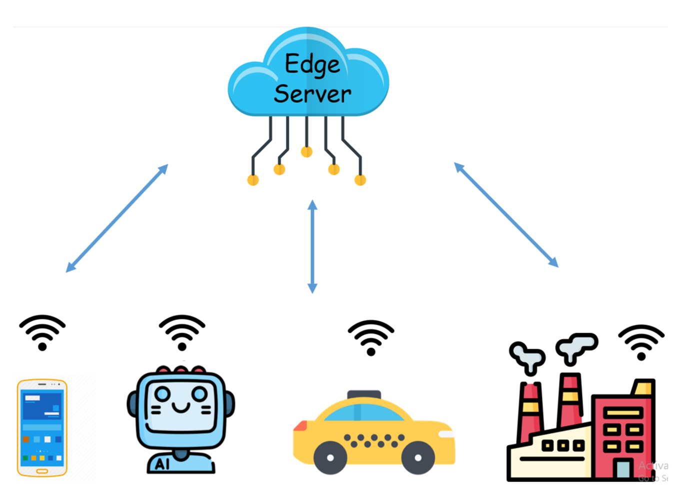

Fig. 1: 2-tier edge computing overview

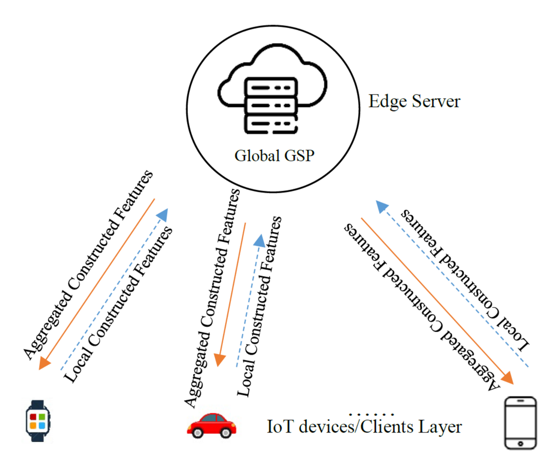

Fig. 2: Federated Feature Construction Procedure

Proposed Method

- The first federated feature construction (FFC) method called multimodal multiple FFC (MMFFC) is proposed, where IoT devices collaborate to construct multiple informative features without sharing their local datasets.
- The proposed FFC algorithm uses multimodal optimization and gravitational search programming (GSP) in a federated and collaborative manner to construct multiple high-level features and provide an improved trade-off between communication cost and learning accuracy.
- Inspired by original version of federated learning (FL) algorithm to train the global GSP in an edge server, the local GSP should be executed in clients and construct multiple features from their datasets by using crowding clustering method.
- Then global GSP continues the process in an edge server to aggregate the constructed features.
- Training of a global GSP 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 Crowding clustering method

Input: The number of programs in the population (S), Number of niches (A), and niche size (NS)
Output: Desired number of niches

Initialize a reference point R randomly in the population $s \leftarrow S$ // s is the number of unclustered programs
for $i = 1 : A$ **do**
  if $s > NS$ **then**
    $ns \leftarrow NS$
  else
    $ns \leftarrow s$ // ns is the niche size for the particular niche
  end if
  Find the closest program Z to R in population
  Fine $ns - 1$ closest programs to Z in population
  Put Z and $ns - 1$ programs in the i-th niche
  Remove the ns selected programs from the population
  $s \leftarrow s - ns$
end for

Algorithm 2 Pseudo code of the global phase at edge server

Input: Multiple local "Best" programs, their corresponding fitness values, and their indices in the local population from all clients
Output: Multiple "Best" solutions (Best constructed features)

while reaching the stopping criteria or maximum iteration **do**
  for all clients **do**
    Select NS randomly and compute A
    Execute Algorithm 3
    Receive multiple best programs, their corresponding fitness values and indices
  end for
  Global population = all local best programs
  Global fitness = corresponding fitness values of local best programs
  Updating Kbest, G, Best, and M based on the GSP
  Calculating global programs' acceleration and velocity
  Updating global programs' position
  Send the updated programs to their clients
end while

Algorithm 3 Pseudo code of the local phase at client

Input: The number of programs, Number of features, Depth of programs, Number of operands, A, and the updated "Best" programs of the client from the global algorithm
Output: Multiple local "Best" programs and their corresponding fitness values, and indices

Initialization: Initial a population of fixed-length strings randomly
In iterations > 1, use saved population from last iteration and replace the updated "Best" programs
Use Algorithm 1 to cluster the population
while reaching the stopping criteria/maximum iteration **do**
  Evaluate programs by IGR (2)
  Updating Kbest, G, Best, and M for each niche (GSP)
  Calculating the acceleration and velocity of each niche population (GSP)
  Updating programs' position of each niche (GSP)
end while
save the position, velocity, and fitness of local programs

Data and Results

- The proposed method is evaluated through two scenarios. 1) MMFFC federated learning and 2) MMFFC centralized learning. For the first and second scenarios, IoT network datasets and UCI datasets are used, respectively.
- The details of these datasets are provided in Table I and Table II.

TABLE I: Characteristics of the IoT datasets

Dataset name	Classes	Features	Instances
DEFT	16	111	7289
ACC	2	30	284807
KDD99	5	41	494021

TABLE II: Characteristics of the UCI datasets

Dataset name	Classes	Features	Instances
Wine	3	13	178
Sonar	2	60	208
Wdbc	2	30	569
Hill Valley	2	100	606
Ionosphere	2	34	351
Balance-Scale	3	4	625
Iris	3	4	150
Thyroid	3	5	215

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

$$CA = \frac{\#\,correctly\ classified\ instances}{\#\,total\ instances} * 100$$

$$FR = \frac{\#\,total\ features\ - \#\,constructed\ features}{\#\,total\ features} * 100$$

TABLE III: Comparison between the proposed method and No-FS and four other methods in the literature

Dataset	Method	(#) Features	FR (%)	CA (%)	Dataset	Method	(#) Features	FR (%)	CA (%)
DEFT (iid)	No-FS	111	-	95.25	DEFT (non-iid)	No-FS	111	-	94.73
	MMFFC	21	81.08	**96.13**		MMFFC	17	84.68	**94.83**
	Fed-FS-GSA [11]	63	43.24	94.91		Fed-FS-GSA [11]	57	48.64	93.55
	Fed-FS-CE [5]	11	90.09	29.57		Fed-FS-CE [5]	9	91.89	28.02
	MFPSO [10]	48	55.48	82.79		MFPSO [10]	49	55.85	82.38
	FSHFL [12]	87	21.62	95.38		FSHFL [12]	84	24.32	94.46
ACC (iid)	No-FS	30	-	96.41	ACC (non-iid)	No-FS	30	-	94.59
	MMFFC	12	60.00	**98.02**		MMFFC	9	70.00	**96.84**
	Fed-FS-GSA [11]	17	43.33	95.39		Fed-FS-GSA [11]	21	30.00	93.11
	Fed-FS-CE [5]	6	80.00	37.68		Fed-FS-CE [5]	5	83.33	33.61
	MFPSO [10]	13	56.66	80.65		MFPSO [10]	12	60.00	79.24
	FSHFL [12]	26	13.33	96.41		FSHFL [12]	26	13.33	95.60
KDD99 (iid)	No-FS	41	-	99.74	KDD99 (non-iid)	No-FS	41	-	99.50
	MMFFC	23	43.9	99.01		MMFFC	22	46.34	98.32
	Fed-FS-GSA [11]	25	39.02	97.14		Fed-FS-GSA [11]	26	36.58	96.93
	Fed-FS-CE [5]	11	73.17	24.36		Fed-FS-CE [5]	8	80.48	21.08
	MFPSO [10]	20	51.21	92.52		MFPSO [10]	18	56.09	92.19
	FSHFL [12]	20	51.21	**99.82**		FSHFL [12]	20	51.21	**99.64**

TABLE IV: Comparison between the proposed method and two existing centralized FC methods in the literature

Dataset	Method	(#) Features	FR (%)	CA (%)
Wine	No-FS	13	-	83.96
	MMFFC	4	69.23	**97.36**
	Fcm [15]	10	23.07	92.78
	FCMFS [7]	4.3	66.92	91.94
Sonar	No-FS	60	-	68.72
	MMFFC	10	83.33	**92.45**
	Fcm [15]	10	83.33	71.72
	FCMFS [7]	5.5	90.83	72.61
wdbc	No-FS	30	-	92.22
	MMFFC	8	73.33	**97.72**
	Fcm [15]	10	66.66	95.62
	FCMFS [7]	5.03	83.23	95.75
HillValley	No-FS	100	-	50.38
	MMFFC	23	77.00	**99.55**
	Fcm [15]	10	90.00	99.33
	FCMFS [7]	3.28	96.72	99.00
Ionosphere	No-FS	34	-	86.27
	MMFFC	12	64.70	**92.38**
	Fcm [15]	10	70.58	90.65
	FCMFS [7]	4.60	86.47	89.54
Balance-Scale	No-FS	4	-	77.07
	MMFFC	3	25.00	98.39
	Fcm [15]	10	-	98.67
	FCMFS [7]	2.10	47.5	**99.26**
Iris	No-FS	4	-	93.48
	MMFFC	3	25.00	**96.67**
	Fcm [15]	10	-	93.11
	FCMFS [7]	4.63	-	92.30
Thyroid	No-FS	5	-	89.37
	MMFFC	3	40.00	**95.56**
	Fcm [15]	10	-	94.55
	FCMFS [7]	4.47	10.6	94.02

Conclusion and Future Work

- In this work, a multimodal multiple federated feature construction method with gravitational search programming is proposed for the first time.
- The experimental results on three 3 IoT datasets and 8 UCI datasets demonstrate that the proposed MMFFC beneficial for constructing informative features.
- For example, in ACC dataset, the FR of the proposed method is 60% and its accuracy is 98.02 which is increased about 1.6% compared to the accuracy of the classifier with no feature selection/construction.
- In this work, we use crowding clustering strategy and combine it with FFC for the first time.
- We can explore other multimodal strategies such as speciation and fitness sharing in combination with federated feature selection (FFS) and FFC.

References

1. A. Mahanipour and H. Khamfroush, "Multimodal Multiple Federated Feature Construction Method for IoT Environments," in 2023 Global Communication (GLOBECOM). IEEE, 2023.
2. X. Zhang, A. Mavromatics, A. Vafeas, R. Nejabati, and D. Simeonidou, "Federated feature selection for horizontal federated learning in iot networks," IEEE Internet of Things Journal, 2023.