Carnegie Mellon University

Privately Predicting Child Developmental Delays using Federated Learning

14828 Engineering Privacy in Software

Final presentation

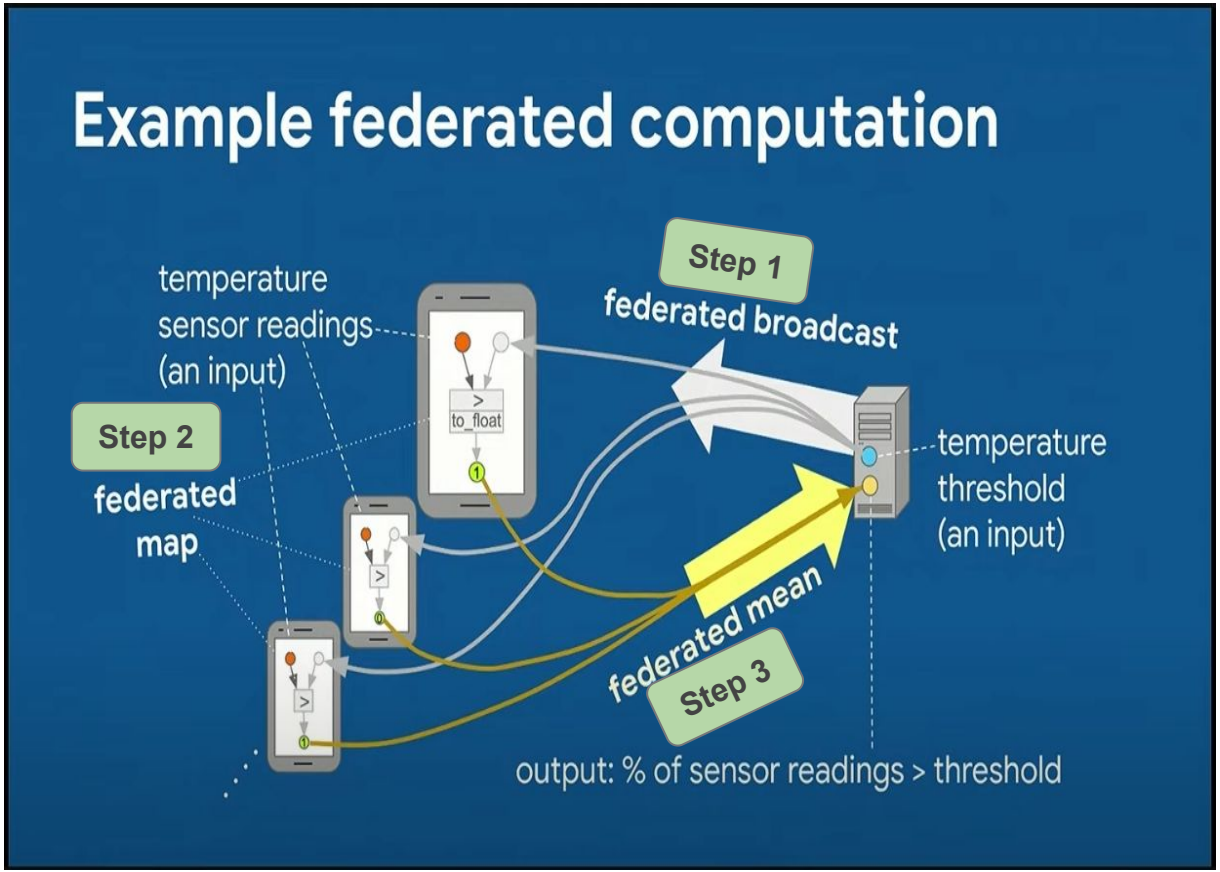
Team 2 : Geetika Gopi, Yunyi Wang, David Meringabo, Kay Chen

Project Summary

- Developmental delays can be measured using a child's overall physical, cognitive, and socio-emotional development.
- Our goal is to develop a machine learning model that predicts child developmental delays in a privacy preserving way using federated learning.[3]
- On a larger scale, this would be deployed as an app available to parents that would collect the child's data and return development results using the FL model

RQ1: How effective is federated learning as compared to centralized learning when predicting child development delays based on real world data?

Carnegie Mellon University



What is a Federated Computation?

Carnegie Mellon University

*Figure from TensorFlow Federated Tutorial Session. GoogleTalks, Youtube. 2020/7/31. https://www.youtube.com/watch?v=JBNas6Yd30A&t=2491s

Privacy risks associated with DLD diagnosis

Surveillance[4]

Interrogation[4]

Identification[4]

Secondary Use[4]

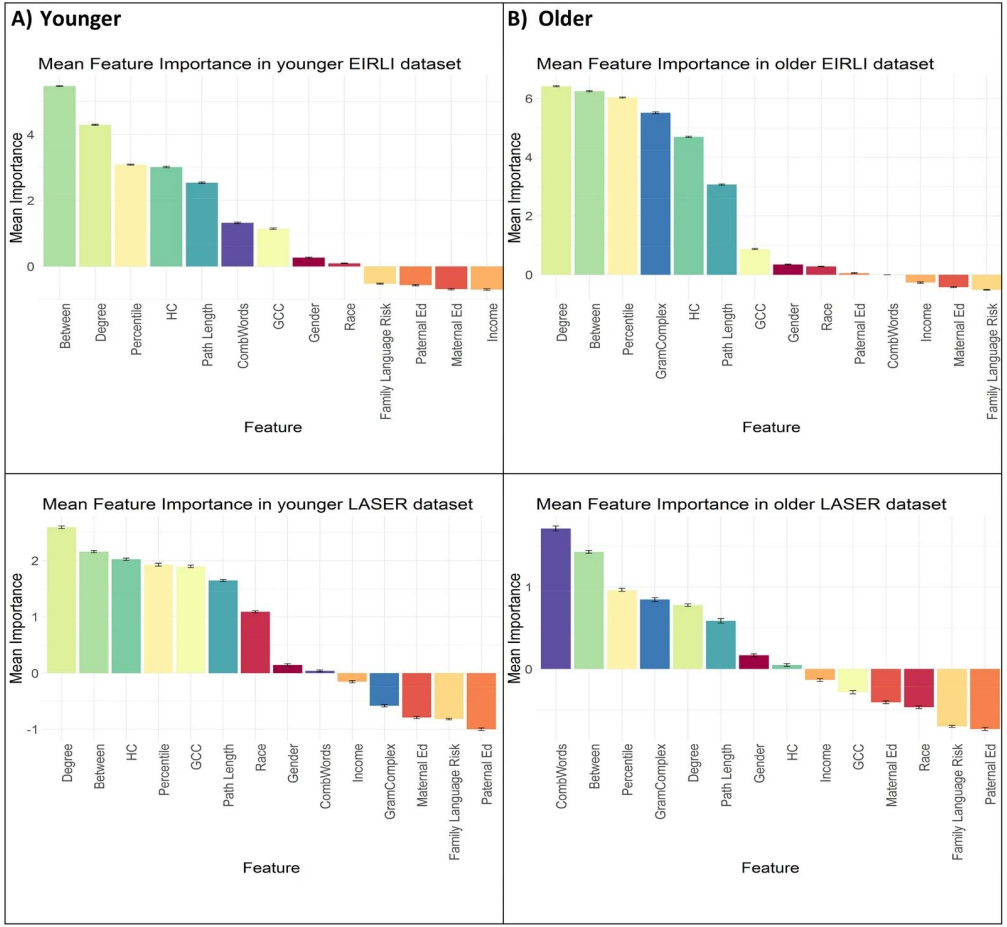
Intrusion[4]

Carnegie Mellon University

Dataset Description

- Datasets
  - MBCDI-derived measures of early vocabulary skills and demographic variables along with outcome measures of later language/reading delays
  - **EIRLI** - Early Identification of Risk for Language Impairment
  - **LASER** - Language Acquisition and Semantic Relations
- **14 input features** including network measures of semantic structure in each child's early productive lexicon, overall vocabulary skill, grammatical ability, and demographic measures[1]
- Binary outcome measure 'AnyLangOrReadDxOnly'
  - **NoDx** - no disorder
  - **Dx** - disorder exists

Carnegie Mellon University



Feature importance for EIRLI and LASER Datasets

Carnegie Mellon University

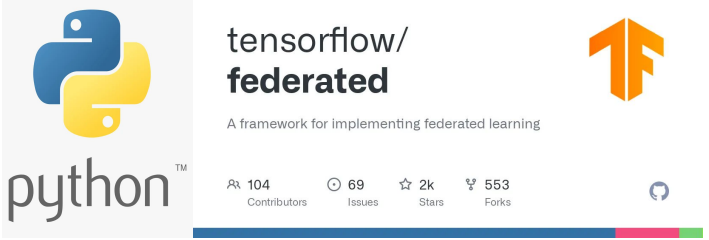
Training a Centralized Learning Model

- Combine all four datasets - (EIRLI_younger.csv, EIRLI_older.csv, LASER_younger.csv, LASER_older.csv)
- Data cleaning and preprocessing
  - Fill in missing values (NaNs) with mean averages
- Split data into training and testing sets (80/20)
- Build a sequential neural network using TensorFlow's Keras API
  - Layer 1 - fully connected, dense, 64 units, ReLU activation function, input shape (14,)
  - Layer 2 - fully connected, dense, 64 units, ReLU activation function
  - Layer 3 - fully connected dense, 1 unit, sigmoid activation function, output between 0 and 1
- Compile the model using adam optimizer and binary cross-entropy loss function
- Train the model using a batch size of 32 and 10 epochs
- Evaluate the model against test dataset

Carnegie Mellon University

Simulating a Federated Learning Model

Step 1: Create a TFF Model



Creating a Keras Sequential model.

Wrap the Sequential model into a TFF model.

```
def model_fn():
  keras_model = create_compiled_keras_model()
  return tff.learning.models.from_keras_model(
      keras_model,
      input_spec=federated_train_data[0].element_spec,
      loss=tf.keras.losses.BinaryCrossentropy(),
      metrics=[tf.keras.metrics.BinaryAccuracy()])
```

Carnegie Mellon University

Simulating a Federated Learning Model

Step 2: Create a training process and run rounds of Federated Computations.

Create a tff training process.

```
training_process = tff.learning.algorithms.build_weighted_fed_avg(
    model_fn,
    client_optimizer_fn=lambda: tf.keras.optimizers.Adam(learning_rate= 0.001),
    server_optimizer_fn=lambda: tf.keras.optimizers.Adam(learning_rate= 0.001))
```

Number of Federated Computation rounds to run (same as Epoch)

Running one round of federated computation.

```
NUM_ROUNDS = 10

for round_num in range(NUM_ROUNDS):
  result = training_process.next(train_state, federated_train_data)
  train_state = result.state
  train_metrics = result.metrics
  print('round {:2d}, metrics={}'.format(round_num, train_metrics))
```

Carnegie Mellon University

Simulating a Federated Learning Model

Step 3: Evaluate the model using test data **[2]**

Builds a federated evaluation process

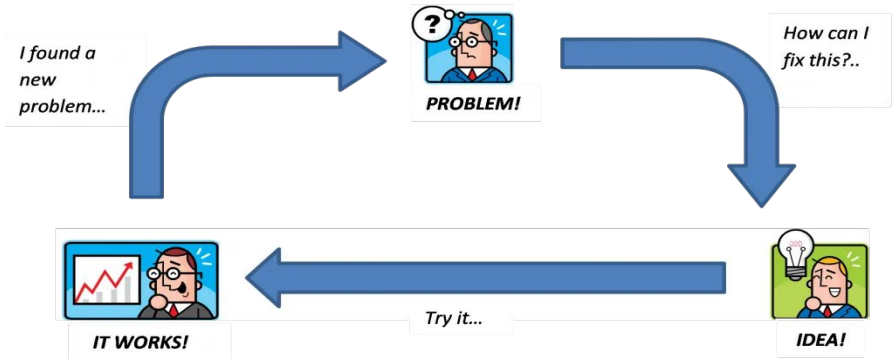
```
evaluation_process = tff.learning.algorithms.build_fed_eval(model_fn)

evaluation_state = evaluation_process.initialize()
model_weights = training_process.get_model_weights(train_state)
evaluation_state = evaluation_process.set_model_weights(evaluation_state, model_weights)

evaluation_output = evaluation_process.next(evaluation_state, federated_test_data)
```

Similar to training rounds, we run an evaluation round.

Carnegie Mellon University

Challenges Faced and Lessons Learnt

- **No** documentation on TFF applied to Random Forest models
- Mainly suited for Keras implementations of ML models (but can be customized for other libraries like Scikit)
- Model needs to be Sequential and TensorFlow Decision Forests is not!



Carnegie Mellon University

Model Baseline to Compare Performance

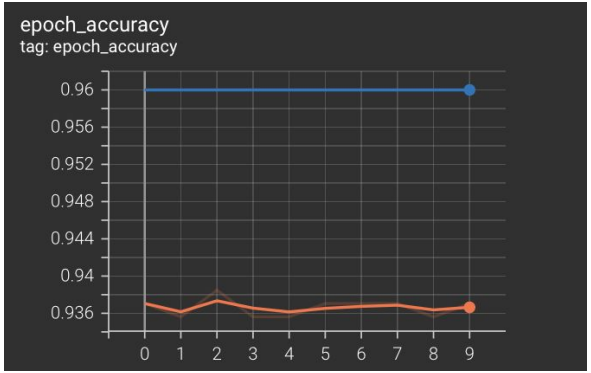
Parameter	Centralized Model	Federated Model
Training and test data split	`test_size=0.2` `random_size=42`	`test_size=0.2` **(To be validated)**

Neural network model and definition	`tensorflow.keras.sequential`
	`Dense(64, activation='relu'`
	`Dense(32, activation='relu')`
	`Dense(1, activation='sigmoid')`
		`tensorflow.keras.sequential`
		`Dense(64, activation='relu'`
		`Dense(32, activation='relu')`
		`Dense(1, activation='sigmoid')`

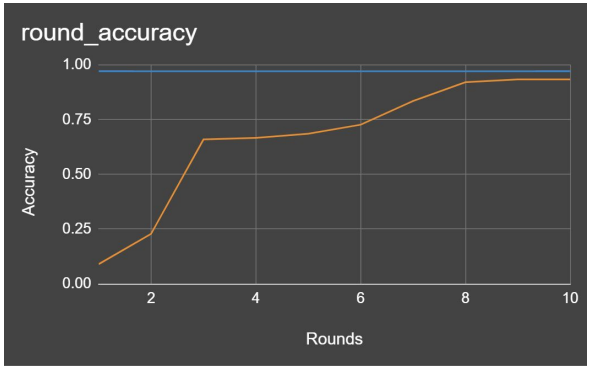
Model compile parameters	`optimizer='adam',`
	`loss='binary_crossentropy',`
	`metrics=['accuracy']`
		`optimizer='adam',`
		`loss='binary_crossentropy',`
		`metrics=['accuracy']`

Validation dataset	`X_test, y_test`	**(To be validated)**

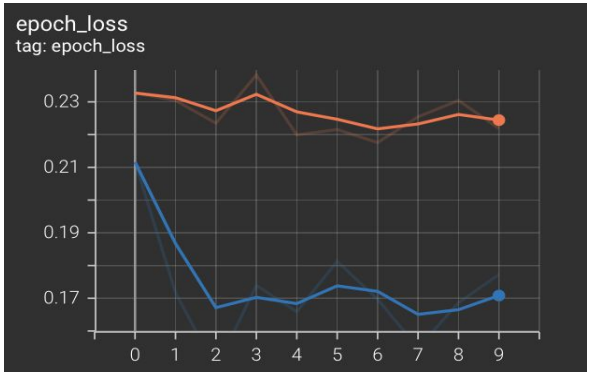
Preliminary Results: Centralized vs Federated Learning



Centralized Learning Accuracy: 0.960



Federated Learning Accuracy: 0.97126



Centralized Learning Loss: 0.177

Federated Learning Loss: 0.40975

Carnegie Mellon University

Next Steps

- Explore confusion matrices for both centralized and federated learning models
- Attempt to reason out the difference in accuracy and/or loss function
- Complete the final paper

Carnegie Mellon University

References

[1] Moving towards accurate and early prediction of language delay with network science and machine learning approacheshttps://www.nature.com/articles/s41598-021-85982-0

[2] TensorFlow Federated Tutorial, MNIST Image Classification Example, Evaluation Step. https://www.tensorflow.org/federated/tutorials/federated_learning_for_image_classification#evaluation

[3] Wikipedia Federated learning

https://en.wikipedia.org/wiki/Federated_learning#:~:text=Federated%20learning%20(also%20known%20as,each%20using%20its%20own%20dataset.

[4]CITRON, D. K., AND SOLOVE, D. J. Privacy harms. BUL Rev. 102 (2022), 793.

Carnegie Mellon University