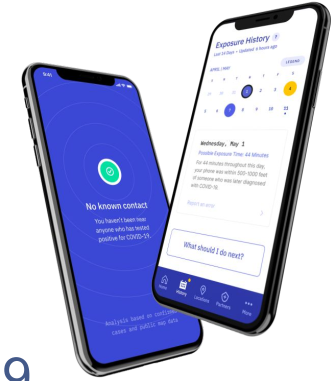
NIST TC4TL Challenge

PathCheck Foundation

Sheshank Shankar, Ayush Chopra, Rishank Kanaparti,
Myungsun Kang, Abhishek Singh, Ramesh Raskar

MIT and PathCheck

PathCheck Foundation, MIT spin-off



- Helping States and Nations launch GAEN apps

  Contracts with **5 US states/territories, 2 Countries**

- World's largest open-source non-profit project for Covid19

  Privacy first solutions for the pandemic and restarting economy

- 20 full time software engineers, 50 FT professional volunteers

  Epidemiologists, Privacy, Legal, Ethicists, Behavior scientists

- Long term philanthropic funding

The New York Times | THE WALL STREET JOURNAL. | npr | WIRED | The Atlantic | MIT Technology Review | U.S. HOUSE OF REPRESENTATIVES

Introduction

- **Challenge:** RSSI Signal Strength of BLE is very noisy.
- **Problem**: Estimate distance between 2 phones given the time series of phone sensor data.
  - Proximity sensing is concerned with predicting if two individuals have been in ”close contact” for ”too long” that may open the *possibility* of COVID-19 transmission.
- **Methods**
  - Deep learning based
    - LSTM
    - GRU
    - ConvGRU
    - Conv1D
    - Feed Forward
  - Support Vector Machine
  - Decision Tree
    - XGboost
    - Random Forest
  - Nearest neighbour
- **Results**
  - Conv1D best results
- **Analysis**
  - Ablation Studies
  - Data Analysis
  - Training and Dev set Discrepancy

Method

Data Processing: Tested Approaches

- Mix-up Data Augmentation
  - Increase effective size of training dataset size
  - No increase in performance

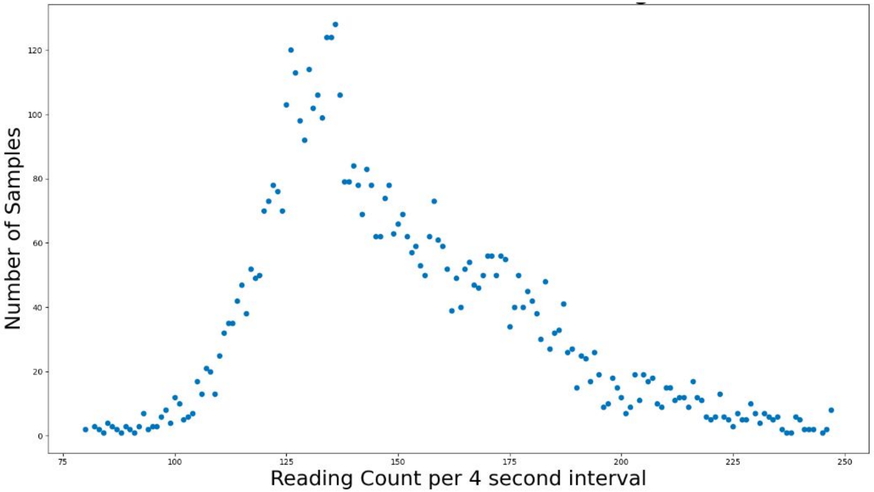
- Nearest k train
  - Subsample training dataset to align distribution with val set [stress test]
  - Limited increase in performance

Data Processing: Current Approach

- Breakdown into 150 time-steps / 4 second interval
  - Minimize need for undersampling and oversampling data points (to mitigate noise)
  - Every time-step represented as normalized fixed-length feature vector

- Metadata is One-Hot Encoded and concatenated for each time-step vector
  - All readings concatenated into single feature vector / 4 second interval*

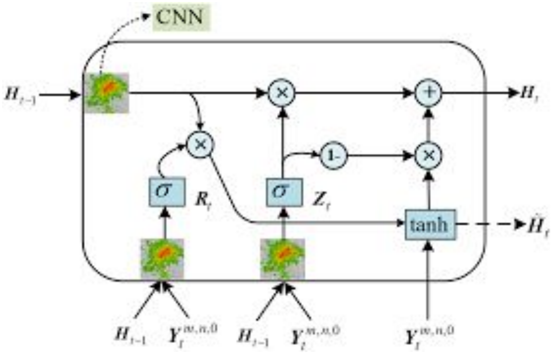
Distribution of # of Sensor Reading Counts



*When model does not use time-series input

Model Architecture: Deep Learning

- **LSTM**
  - Time-Series input format
  - Implementation experiments
    - Multiple Layers
    - Varying Hidden Sizes
- **Temporal Conv1D**
  - Inspiration from Google's Wavenet
  - Wavenet made use of Conv1D neural net for predicting the sequential audio signal.
    - 1D CNN + Dropout
    - 1D CNN + Dropout + Maxpool
    - 1D CNN + Dropout + Dilation
  - Experimented with hyperparameters
- **ConvGRU**
  - GRU with Conv1D reset, update and output gates
  - Implementation experiments:
    - No. of epochs
    - Batch size
    - Weight Decay
    - Learning Rates
- **Feed Forward**
  - Concatenated time-step input format
  - Implementation experiments
    - Multiple Layers
    - Dropout
    - Activation Functions



ConvGRU Architecture Model

Model Architecture: Support Vector Machines and Decision Tree

- Support Vector Machine
  - Concatenated time-step input format
  - Implementation
    - Nu-Support Vector Classification
    - C-Support Vector Classification

- Decision Tree
  - Concatenated time-step input format
  - Implementation
    - XGBoost
    - Random Forest Classification

Results

Results

- Hardware:
  - Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz server (528 GB RAM, 48 cores) on a single GPU
- Implementation:
  - Pytorch
    - Completely trained
  - Scikit-learn
    - Partially trained
- Best Networks:
  - ConvGRU (NIST)
  - Temporal Conv1D (MITRE)

*** *All models optimized using Adam optimizer*

Network Description	Train Set	Train %	Epochs	Batch Size	1.2m FINE	1.8m FINE	3m FINE	1.8m COARSE
GRU	NIST dev	90.0	200	100	0.65	0.13	0.28	0.08
ConvGRU	**NIST dev**	**100.0**	**200**	**100**	**0.37**	**0.04**	**0.23**	**0.02**
ConvGRU	MITRE	100.0	500	4000	1.07	1.0	0.98	1.05
LSTM	MITRE	100.0	40	100	1.0	1.08	0.93	0.97
GRU	MITRE	100.0	40	100	1.02	0.99	0.93	0.97
Feed Forward	MITRE	100	100	500	0.71	0.79	0.85	0.75
Temporal Conv1D	**MITRE**	**100.0**	**100**	**50**	**0.62**	**0.61**	**0.59**	**0.53**
C-SVC	MITRE	1.0	-	100	1.01	0.97	0.97	1.01
Nu-SVC	MITRE	1.0	-	100	0.82	0.8	0.78	0.69
XGBoost	MITRE	2.0	-	100	1.0	1.04	1.03	1.04
Random Forest	MITRE	100.0	-	100	1.0	1.05	1.02	1.1

Analysis

Ablation studies

- Trained with a medley of input data streams, feeding a subset of the data to estimate the sensors which would give us the best results for the TC4TL task, using our training scheme.
- We excluded a few sensors, and trained it on rest of the data, thus requiring minimal adjustment on first layer of our neural-net, and accommodated varying sized input feature vectors
- Performed initial experiments by excluding device-level information (~35%) - *TXDevice, RXDevice*, *TXPower*, *RxPower*, *Device Carriage*, *and Activity*, but didn't observe any major improvements, but rather made it unstable, and susceptible to overfitting on two classes
- Other studies done like -
    - Only bluetooth
    - Only Bluetooth and Gyroscope
    - Only attitude excluded.
- Shall take others like gyroscope (for orientation of Bluetooth antenna), accelerometer (for linear motion) , and magnetometer (for magnetic aberration).

Data Analysis

- We investigate variation across different sensor data.
- We visualized PCA in 2d to identify if there are any visible cluster or patterns in data.

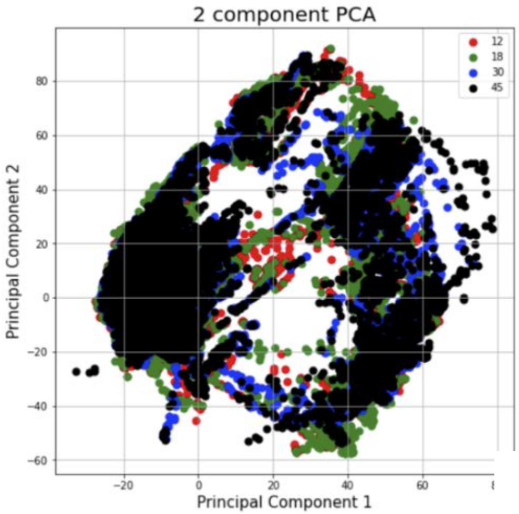


Fig. 2. PCA Visualization of MITRE dataset

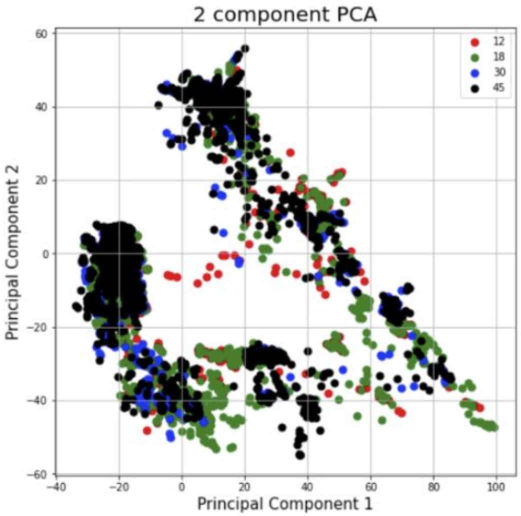


Fig. 3. PCA Visualization of NIST Development dataset

Training and Dev set Discrepancy

- We perform nearest neighbour to analyze the closest points in the training and testing set
- We compare the class-wise distance for the training and dev nearest neighbours.
  - Average l2-norm between closest points: ~12
  - Average l2-norm between closest points with same class: ~200
- We train on a subset of training dataset with 2-nearest neighbour and 1-nearest neighbour with respect to the dev set

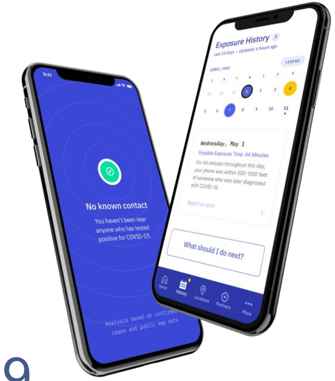
Conclusion

PathCheck Foundation: Conclusion

- Challenges due to the **noise** in the data distribution and **poor transferability** of training data over the validation data.
    - We used only training data and not dev data.
    - The test/dev is too same while training set didn't provide good transferability at all.
    - MITRE set result is close to chance and unclear if any algorithms will be useful in practice
- A physics based model which could capture appropriate invariances will be a good step towards solving the task.
- We also consider interpretable modeling and extensive breakdown of different sensor based data as part of the future work.

Website: pathcheck.org
Email: info@pathcheck.org

PathCheck Foundation, MIT spin-off



- Helping States and Nations launch GAEN apps
  - Contracts with 5 US states/territories, 2 Countries
- World's largest open-source non-profit project for Covid19
  - Privacy first solutions for the pandemic and restarting economy
- 20 full time software engineers, 50 FT professional volunteers
  - Epidemiologists, Privacy, Legal, Ethicists, Behavior scientists
- Long term philanthropic funding