Distributed Random Forests: Resilient?

Raksha Kumaraswany
and
Katherine Metcalf

Problem Description

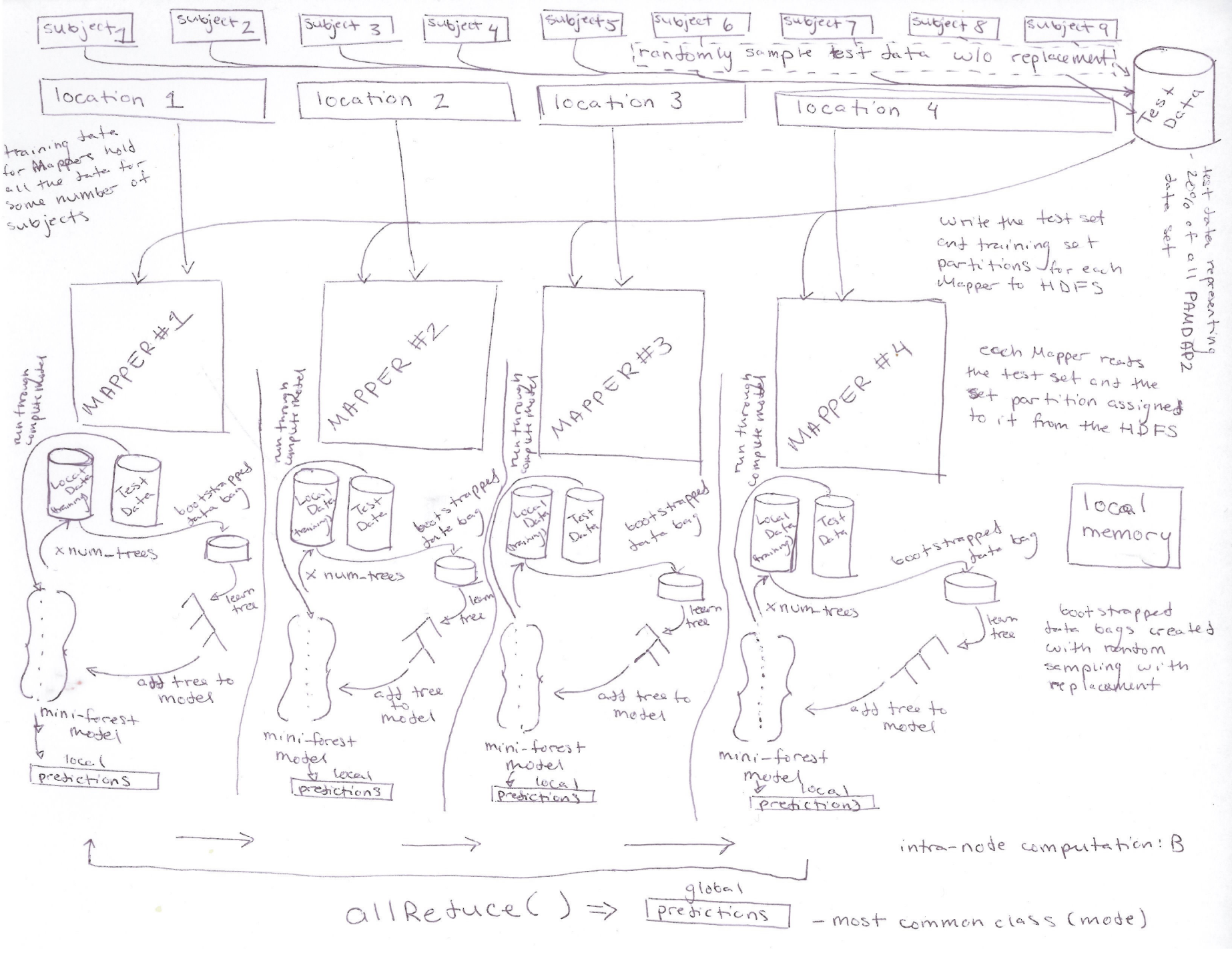
- Explore the effect of learning distributed mini-forests on local data sets that varying according to feature value and label distributions
- Important question b/c passing around data to build partitions with globally representative distributions is computationally expensive and puts the data at higher security risks
- The PAMDAP2 is a time series, activity classification data set:
  - binarized the labels into sedentary and non-sedentary actions
  - 53 continuous feature values from 3 IMU sensors and 1 heart rate monitor
  - > 3M data points
- Split the PAMDAP2 data set among the HARP Mappers by person to create local data sets with distinct feature value and label distributions

Random Forest

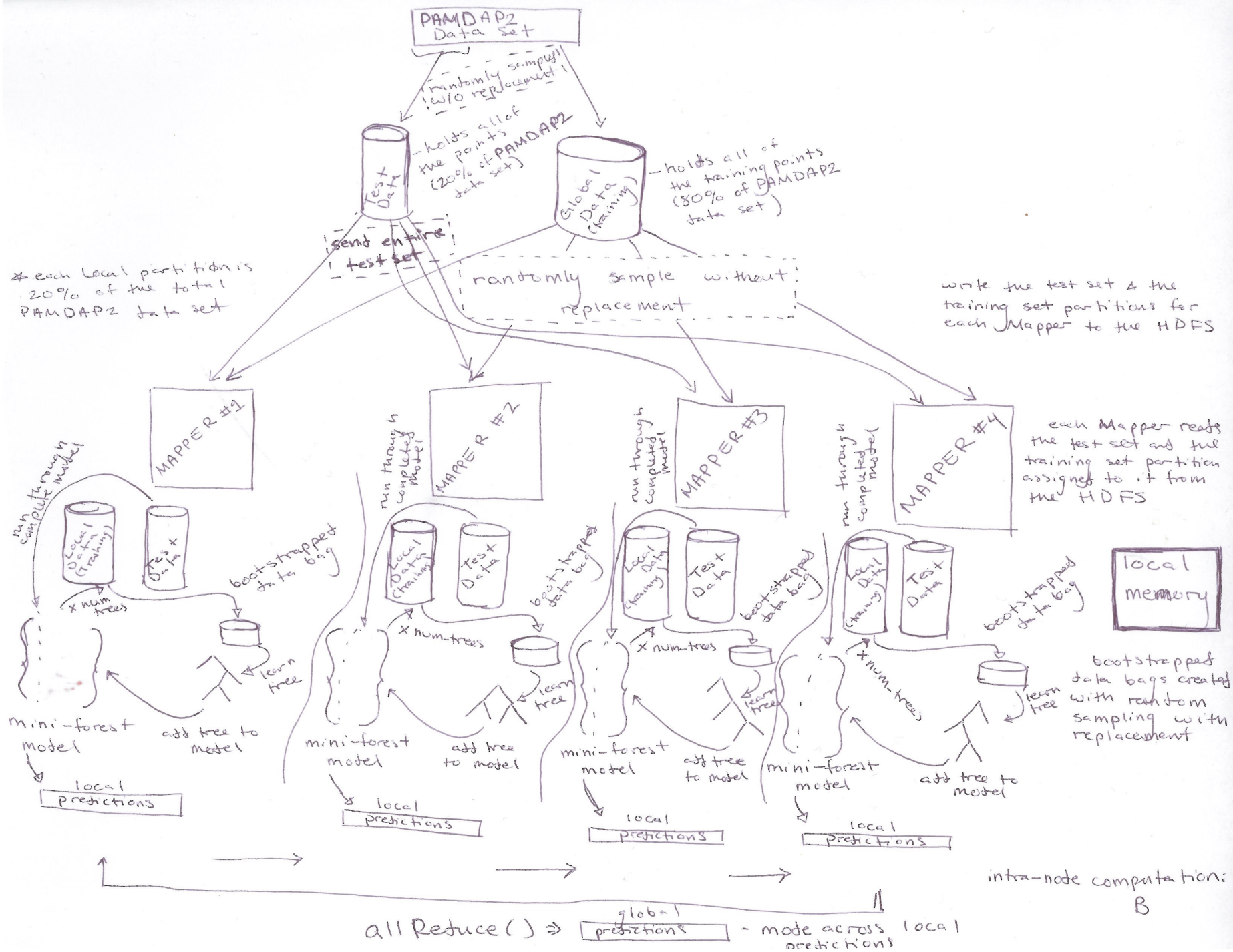
- ensemble classifier
- idea - divide & conquer
- divide - the dataset into bootstrap samples
- conquer - by learning tree for each sample
- gather all trees - a forest, the new model
- prediction - mode, or stochastic prediction based on distribution

- Advantage:
  - these independently learned models, when aggregated create a resilient model
  - not prone to bias
- Disadvantage:
  - computational complexity scaled with data size: many avenues for parallelism
  - bootstrap sample creation, forest creation, tree creation, prediction.
  - prone to variance, but can be controlled with sampling strategies: strategic bag-of-little bootstraps
- What we do - parallel forest creation, with strategic bag-of-little bootstraps

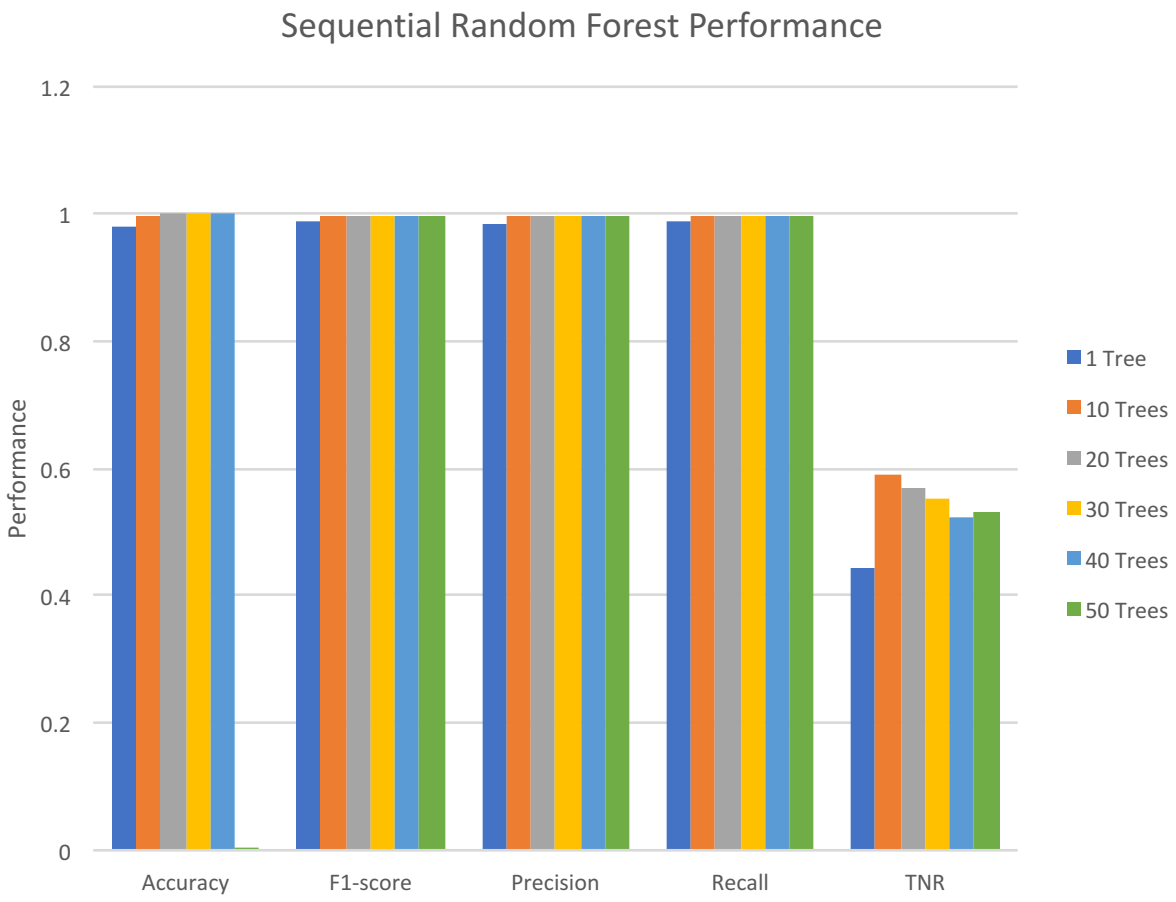
Methods: Local-Label and Local-Label-Feature data

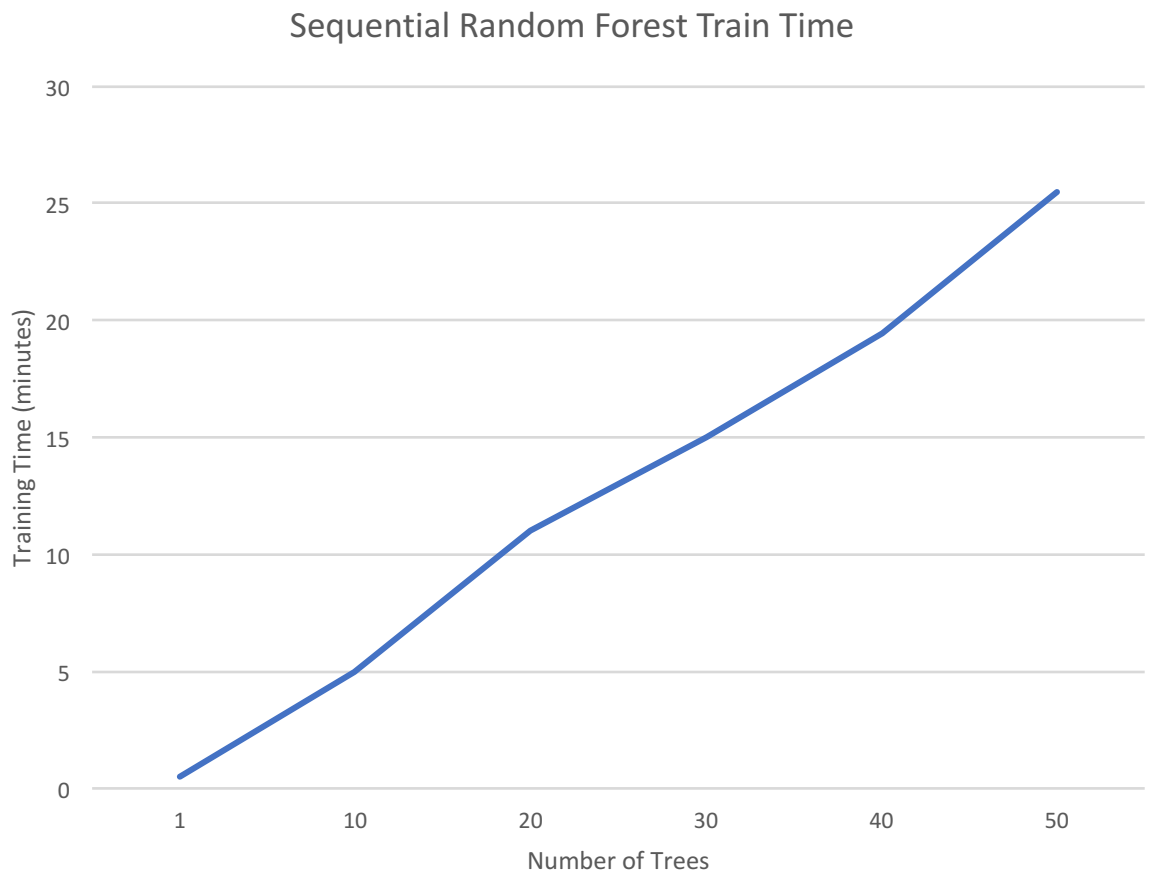


Methods: Global data

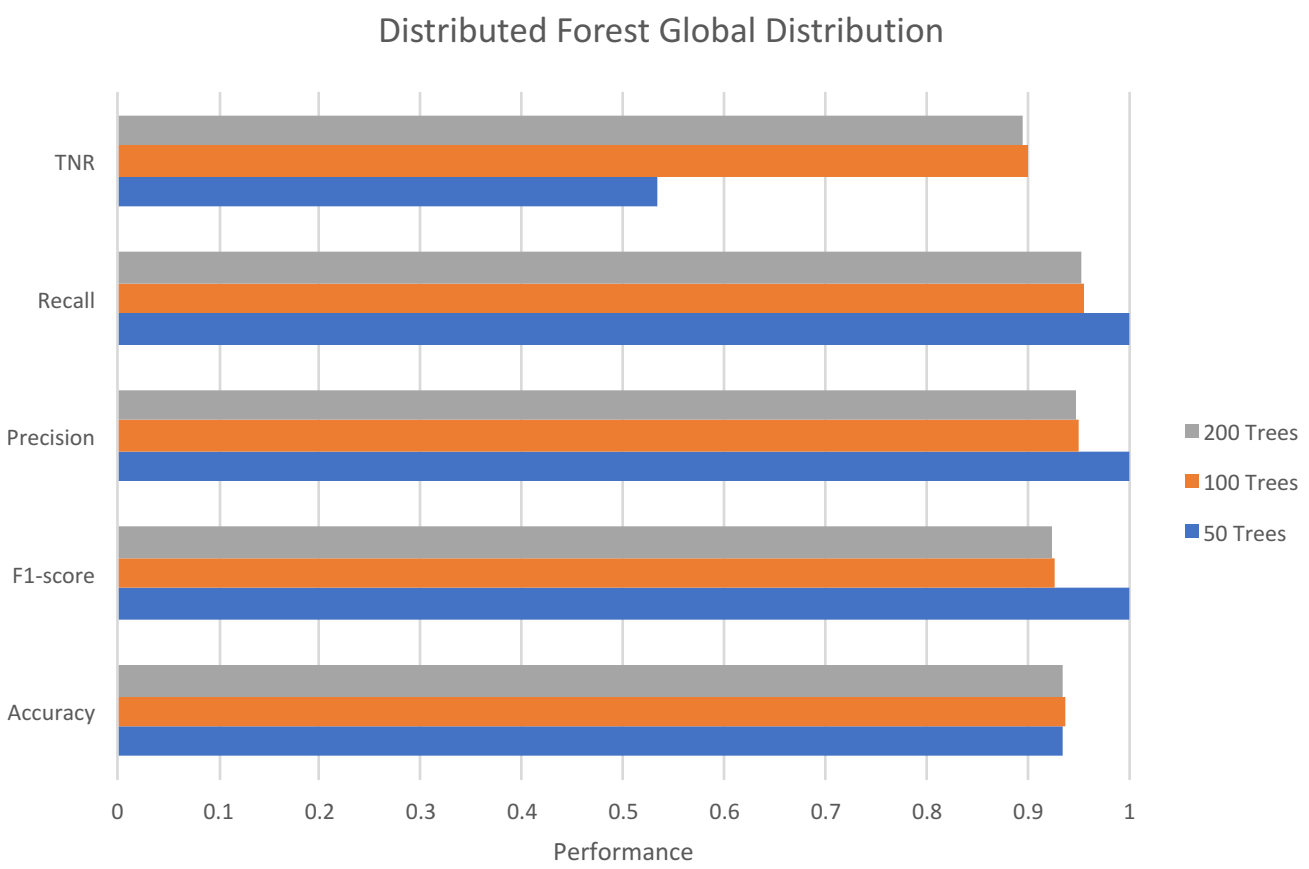


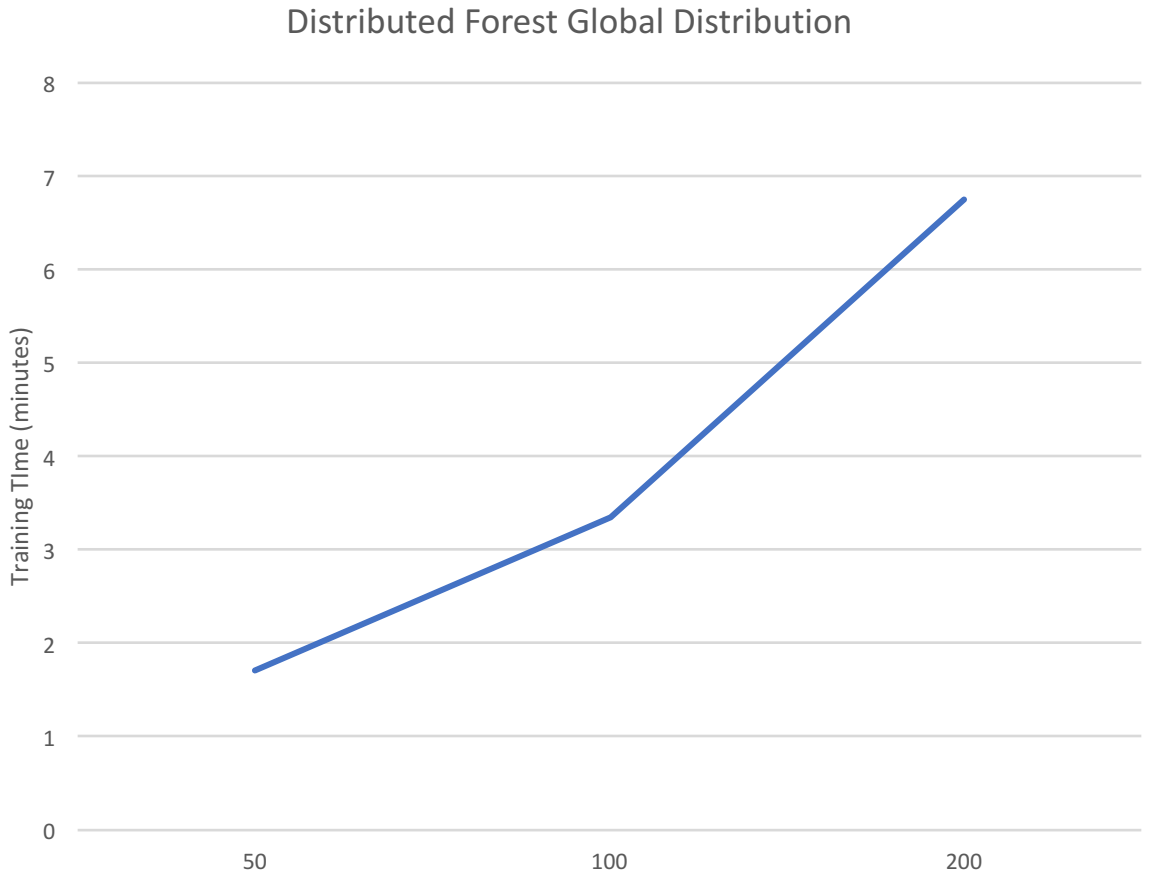
Sequential Results

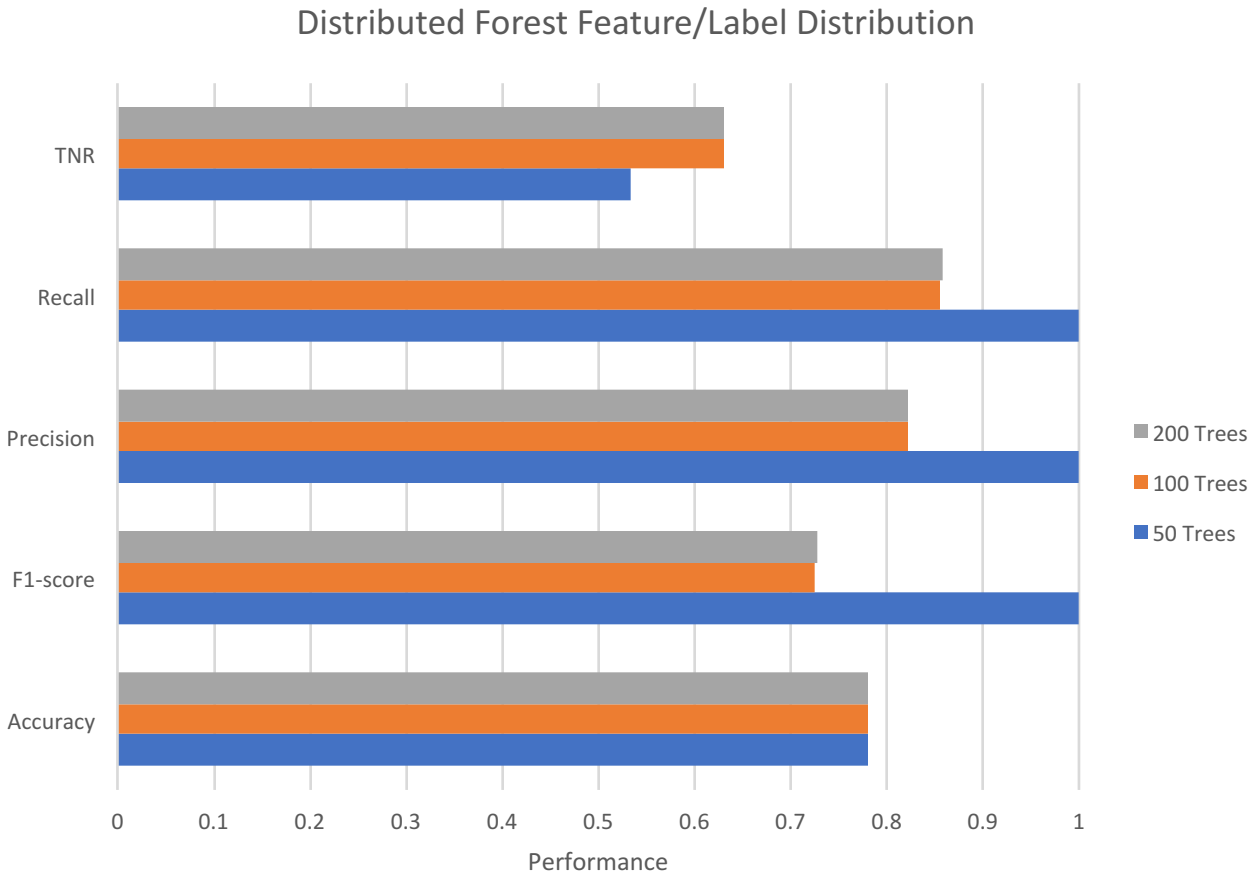


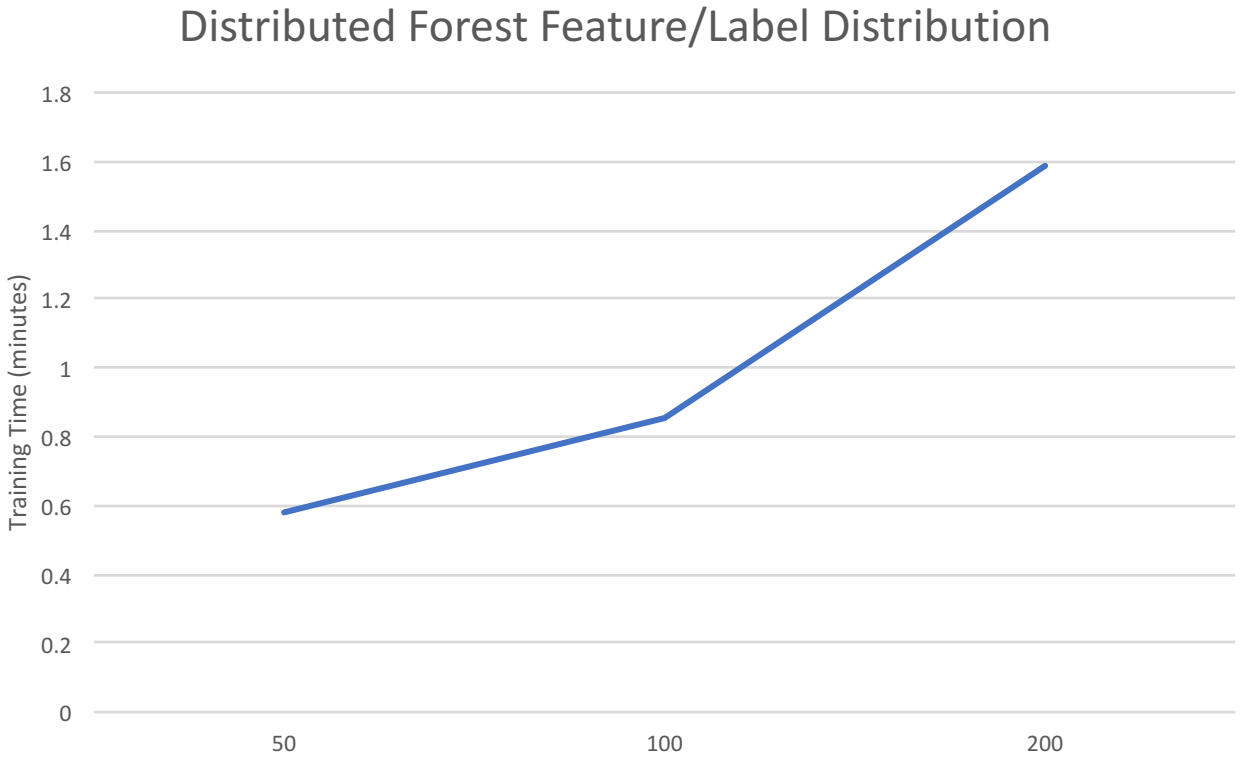


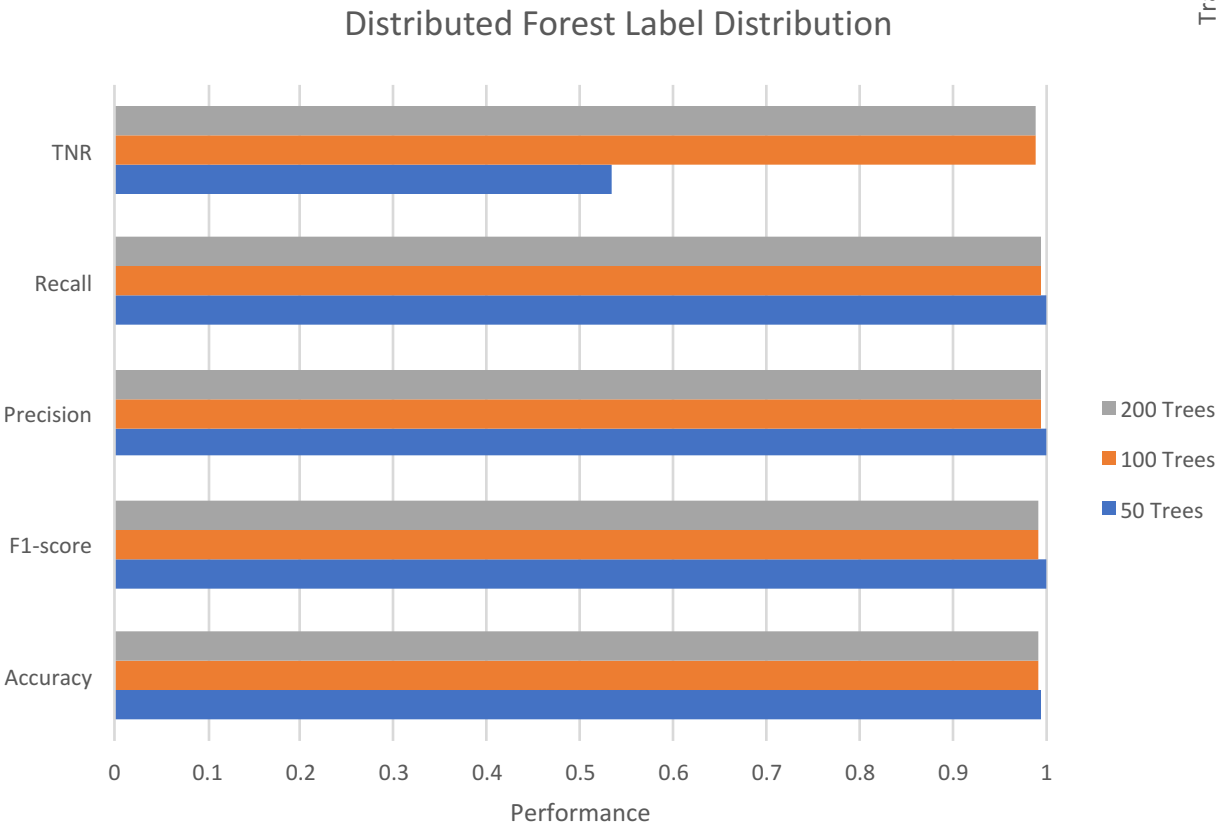
Distributed Results

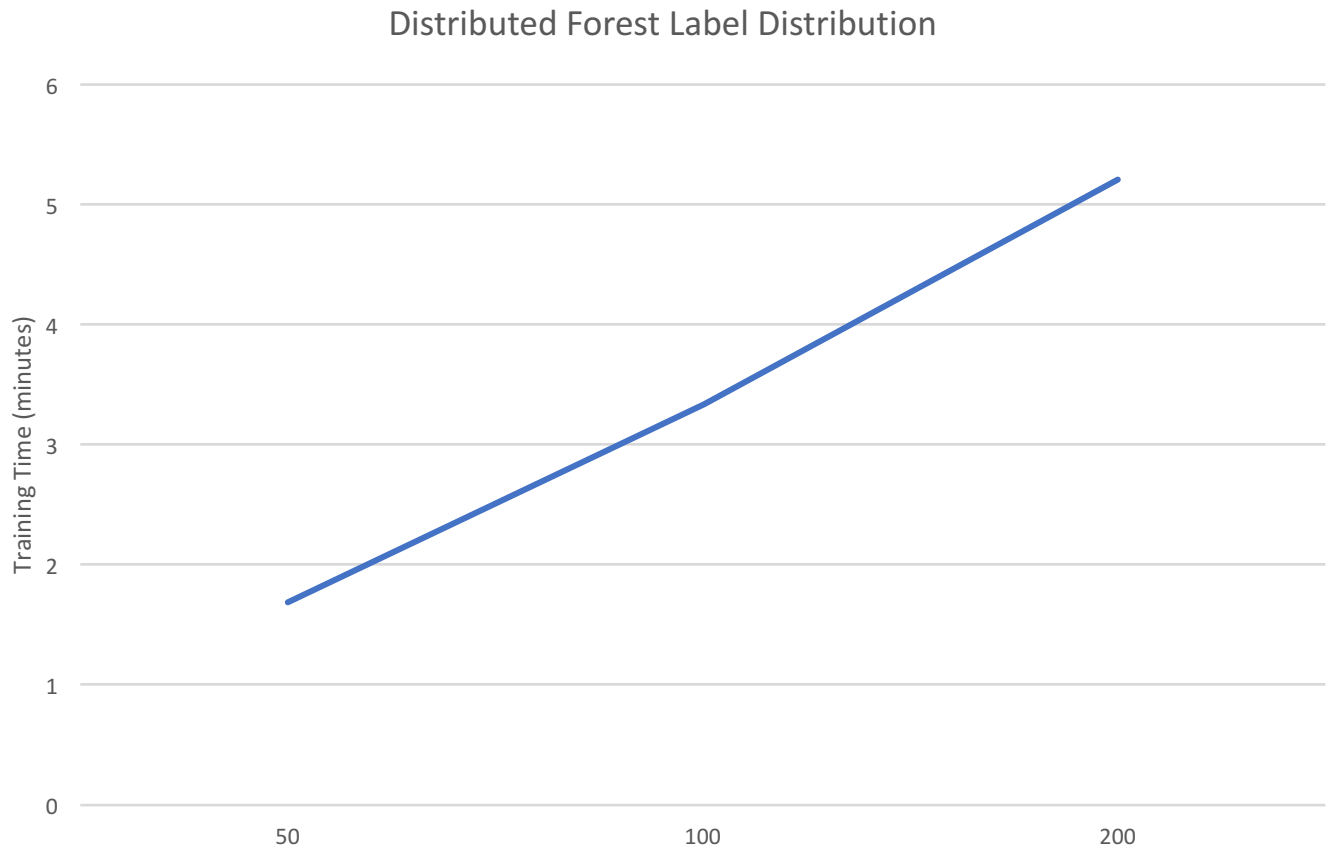





Distributed Forest Feature/Label Distribution
TNR
Recall
Precision
F1-score
Accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Performance
200 Trees
100 Trees
50 Trees


Distributed Forest Feature/Label Distribution
Training Time (minutes)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
50
100
200


Distributed Forest Label Distribution
TNR
Recall
Precision
F1-score
Accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Performance
200 Trees
100 Trees
50 Trees


Distributed Forest Label Distribution
Training Time (minutes)
0
1
2
3
4
5
6
50
100
200

Conclusions

- Random Forest is robust enough to be able to handle large subsets of trees that are built from different label distributions, however, performance drops dramatically when both feature value and label distributions differ
- There is a significant difference in the time is takes to learn the distributed random forests versus the sequential random forests. The differences that can be observed in the results is even more pronounced when you notice that, at most, the sequential code was only run for 50 trees total while the distributed code was run for, 50, 100, and 200 trees per mapper.
- Therefore, this project shows that in cases where the distribution over the feature values does not drastically differ between the data partitions assigned to each mapper, it is not necessary to maintain a global data model to bootstrap from. However, if both the feature value and label distributions differ significantly, then the degradation in performance indicates that it is necessary to take the extra step of maintaining a global data model to bootstrap each mapper's data partition from.

Distributed Random Forests: Resilient?

Raksha Kumaraswany
and
Katherine Metcalf

Problem Description

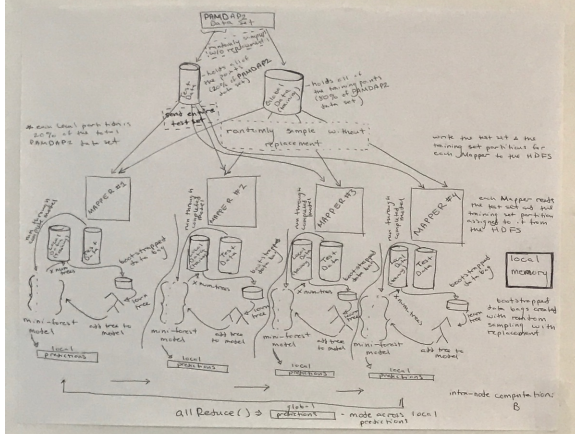
- Explore the effect of learning distributed mini-forests on local data sets that varying according to feature value and label distributions
- Important question b/c passing around data to build partitions with globally representative distributions is computationally expensive and puts the data at higher security risks
- The PAMDAP2 is a time series, activity classification data set:
  - binarized the labels into sedentary and non-sedentary actions
  - 53 continuous feature values from 3 IMU sensors and 1 heart rate monitor
  - > 3M data points
- Split the PAMDAP2 data set among the HARP Mappers by person to create local data sets with distinct feature value and label distributions

Random Fore t

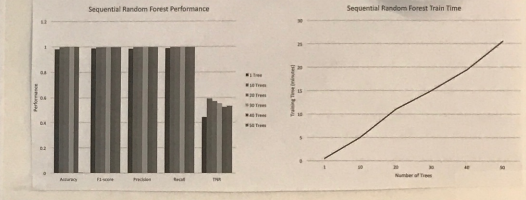
- ensemble classifier
- idea - divide & conquer
- divide - the dataset into bootstrap samples
- conquer - by learning tree for each sample
- gather all trees - a forest, the new model
- prediction - mode, or stochastic prediction based on distribu

- Advantage:
  - these independently learned models, when aggregated create a res
  - not prone to bias
- Disadvantage:
  - computational complexity scaled with data size: many avenues for par
  - bootstrap sample creation, forest creation, tree creation, prediction.
  - prone to variance, but can be controlled with sampling strategies: s
- What we do - parallel forest creation, with strategic bag-of

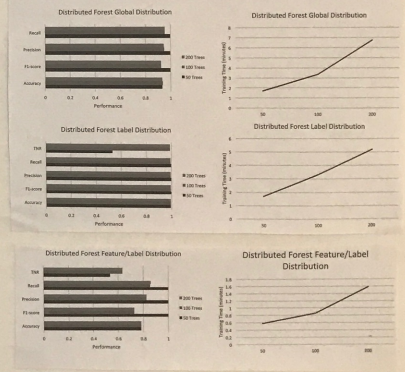
gic bag-of-litt
le bootstraps

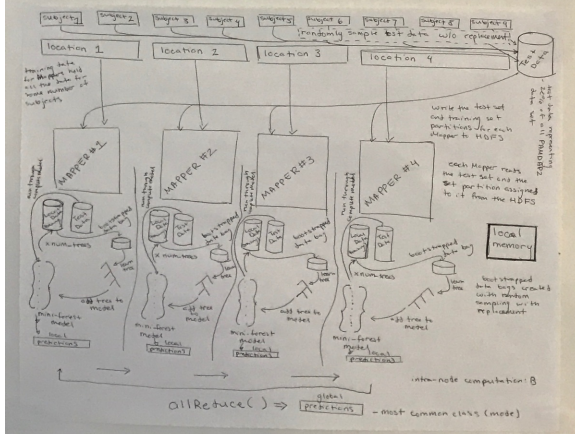


Sequential Results



Distributed Results





Conclusions

- Random Forest is robust enough to be able to handle large subsets of trees that are built from different label distributions, however, performance drops dramatically when both feature value and label distributions differ
- There is a significant difference in the time is takes to learn the distributed random forests versus the sequential random forests. The differences that can be observed in the results is even more pronounced when you notice that, at most, the sequential code was only run for 50 trees total while the distributed code was run for, 50, 100, and 200 trees per mapper.
- Therefore, this project shows that in cases where the distribution over the feature values does not drastically differ between the data partitions assigned to each mapper, it is not necessary to maintain a global data model to bootstrap from. However, if both the feature value and label distributions differ significantly, then the degradation in performance indicates that it is necessary to take the extra step of maintaining a global data model to bootstrap each mapper's data partition from.

Distributed Random Forests: Resilient?

Raksha Kumaraswany
and
Katherine Metcalf

Problem Description

- Explore the effect of learning distributed mini-forests on local data sets that varying according to feature value and label distributions
- Important question b/c passing around data to build partitions with globally representative distributions is computationally expensive and puts the data at higher security risks
- The PAMDAP2 is a time series, activity classification data set:
  - binarized the labels into sedentary and non-sedentary actions
  - 53 continuous feature values from 3 IMU sensors and 1 heart rate monitor
  - > 3M data points
- Split the PAMDAP2 data set among the HARP Mappers by person to create local data sets with distinct feature value and label distributions

Random Fore t

- ensemble classifier
- idea - divide & conquer
- divide - the dataset into bootstrap samples
- conquer - by learning tree for each sample
- gather all trees - a forest, the new model
- prediction - mode, or stochastic prediction based on distribu

- Advantage:
  - these independently learned models, when aggregated create a res
  - not prone to bias
- Disadvantage:
  - computational complexity scaled with data size: many avenues for par
  - bootstrap sample creation, forest creation, tree creation, prediction.
  - prone to variance, but can be controlled with sampling strategies: s
- What we do - parallel forest creation, with strategic bag-of

gic bag-of-litt
le bootstraps

Sequential Results

Distributed Results

Conclusions

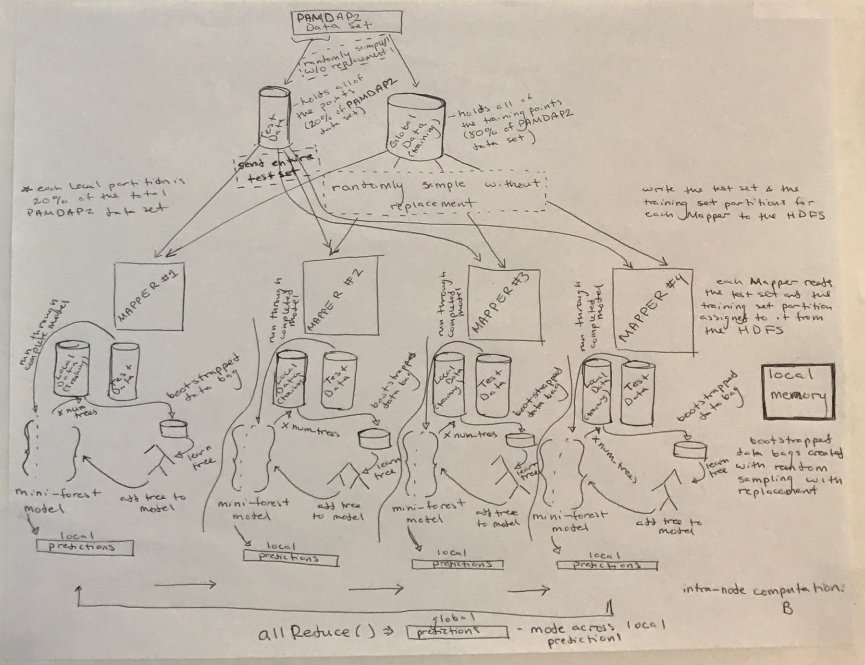
- Random Forest is robust enough to be able to handle large subsets of trees that are built from different label distributions, however, performance drops dramatically when both feature value and label distributions differ
- There is a significant difference in the time is takes to learn the distributed random forests versus the sequential random forests. The differences that can be observed in the results is even more pronounced when you notice that, at most, the sequential code was only run for 50 trees total while the distributed code was run for, 50, 100, and 200 trees per mapper.
- Therefore, this project shows that in cases where the distribution over the feature values does not drastically differ between the data partitions assigned to each mapper, it is not necessary to maintain a global data model to bootstrap from. However, if both the feature value and label distributions differ significantly, then the degradation in performance indicates that it is necessary to take the extra step of maintaining a global data model to bootstrap each mapper's data partition from.

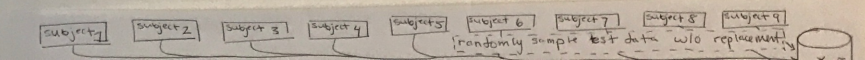
Distributed Random F
Resilient?

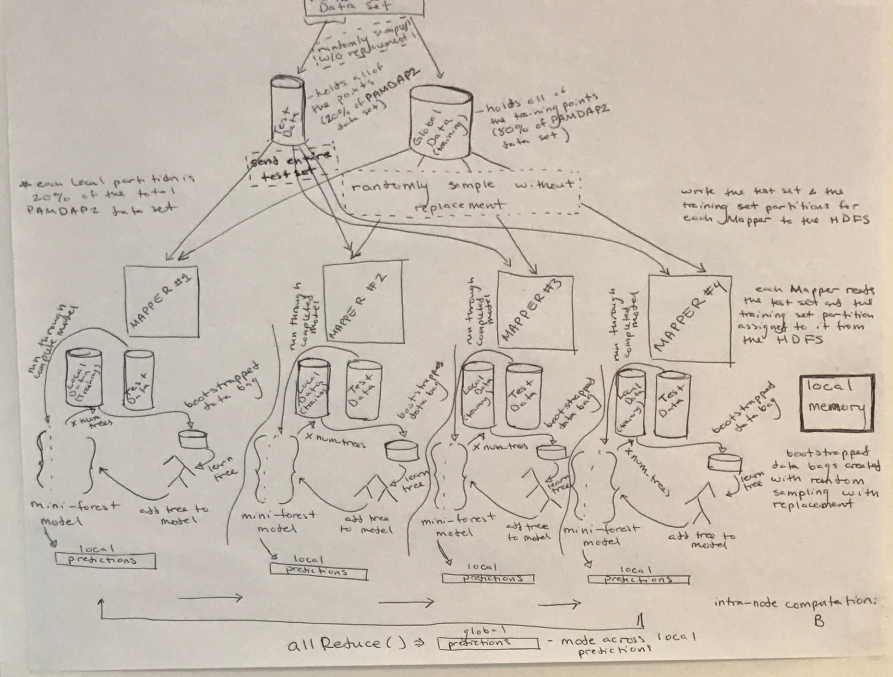
Raksha Kumaraswany
and
Katherine Metcalf

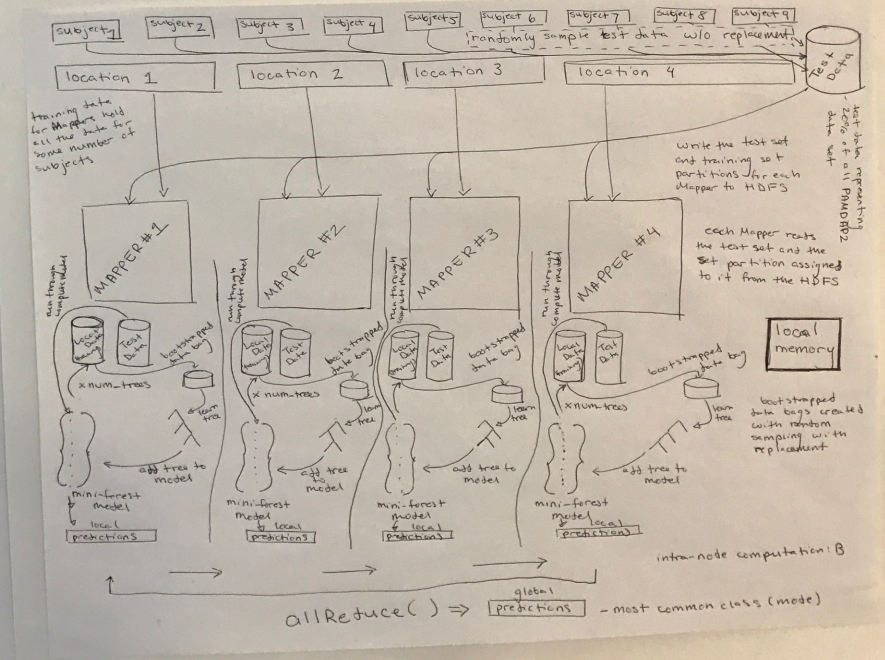
Problem Description

- Explore the effect of learning distributed mini-forests on local data sets that varying according to feature value and label distributions
- Important question b/c passing around data to build partitions with globally representative distributions is computationally expensive and puts the data at higher security risks
- The PAMDAP2 is a time series, activity classification data set:
  - binarized the labels into sedentary and non-sedentary actions
  - 53 continuous feature values from 3 IMU sensors and 1 heart rate monitor
  - > 3M data points
- Split the PAMDAP2 data set among the HARP Mappers by person to create local data sets with distinct feature value and label distributions









nt?

wany

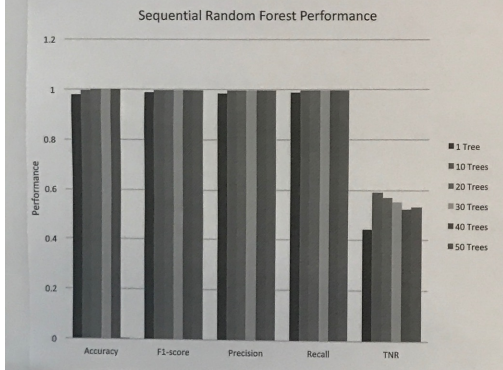
calf

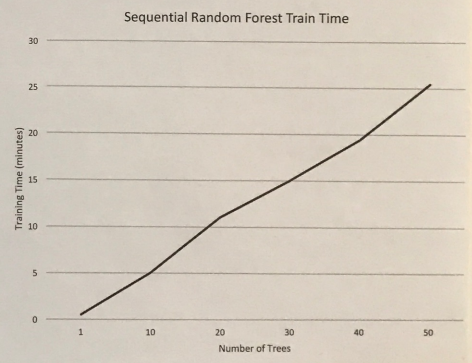
Random Forest

- ensemble classifier
- idea - divide & conquer
- divide - the dataset into bootstrap samples
- conquer - by learning tree for each sample
- gather all trees - a forest, the new model
- prediction - mode, or stochastic prediction based on distribution

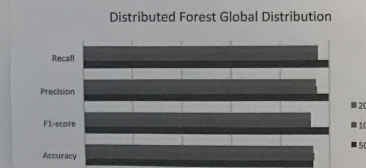
- Advantage:
  - these independently learned models, when aggregated create a resilient model
  - not prone to bias
- Disadvantage:
  - computational complexity scaled with data size: many avenues for parallelism
  - bootstrap sample creation, forest creation, tree creation, prediction.
  - prone to variance, but can be controlled with sampling strategies: strategic bag-of-little bootstraps
- What we do - parallel forest creation, with strategic bag-of-little bootstraps

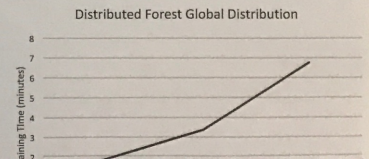
Sequential Results



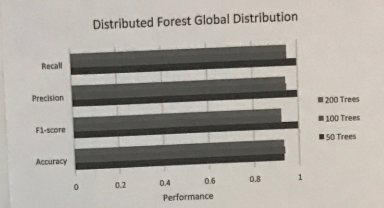


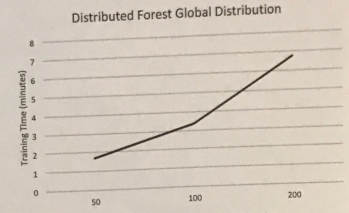
Distributed Results

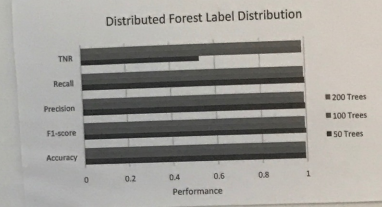


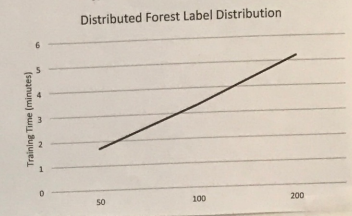


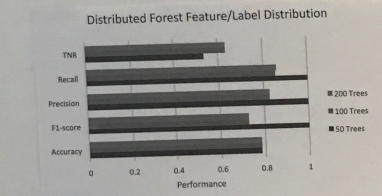
Distributed Results

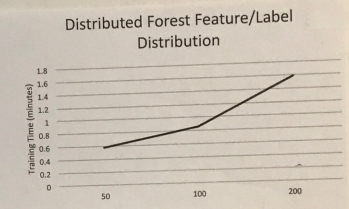












Conclusions

- Random Forest is robust enough to be able to handle large subsets of trees that are built from different label distributions, however, performance drops dramatically when both feature value and label distributions differ
- There is a significant difference in the time is takes to learn the distributed random forests versus the sequential random forests. The differences that can be observed in the results is even more pronounced when you notice that, at most, the sequential code was only run for 50 trees total while the distributed code was run for, 50, 100, and 200 trees per mapper.
- Therefore, this project shows that in cases where the distribution over the feature values does not drastically differ between the data partitions assigned to each mapper, it is not necessary to maintain a global data model to bootstrap from. However, if both the feature value and label distributions differ significantly, then the degradation in performance indicates that it is necessary to take the extra step of maintaining a global data model to bootstrap each mapper's data partition from.