# Performance Analysis of Parallel Support Vector Machines on a MapReduce Architecture

Udita Patel

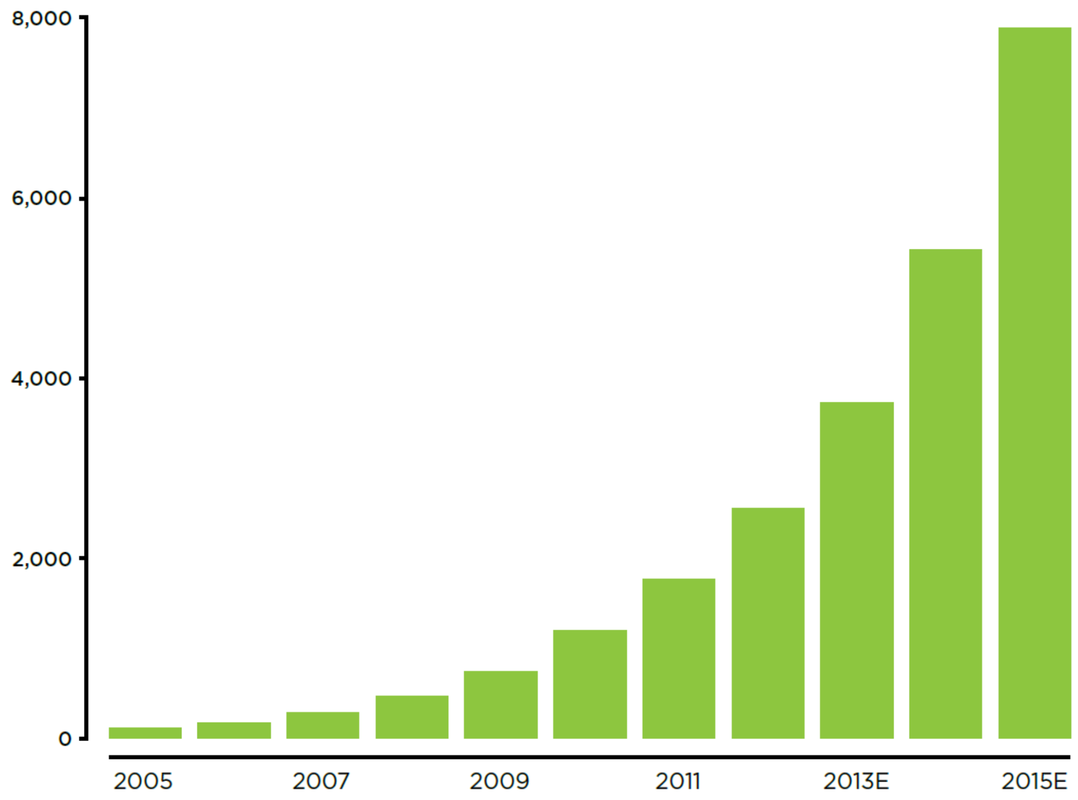
Department of Computer Science

# Agenda

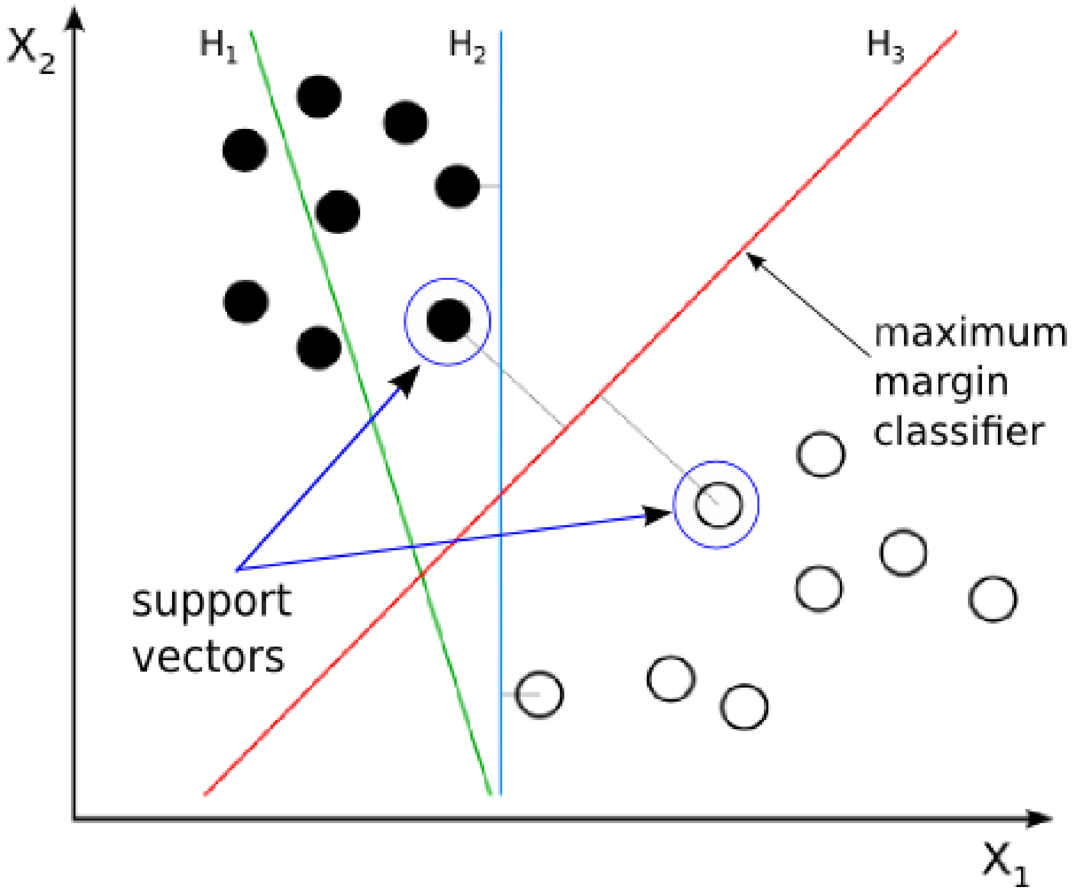
1. Introduction
2. Support Vector Machines
3. Data Preprocessing
4. MapReduce Programming Model
5. Parallel Algorithms
6. Results
7. Conclusions
8. Future Works
9. Acknowledgement

# Introduction

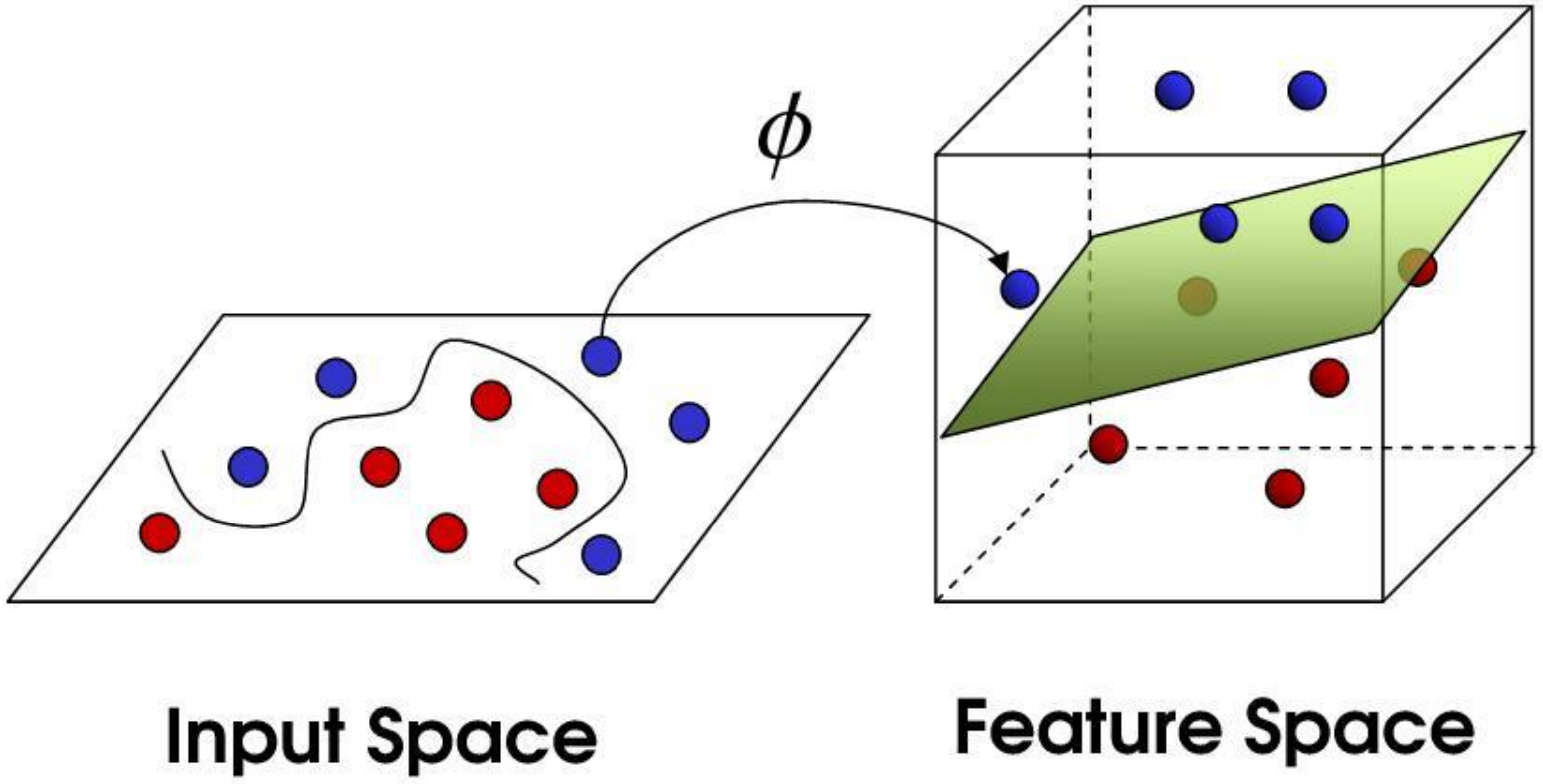
- Support Vector Machines are powerful but computational complexity increases rapidly as the number of training example increases.
- Many Parallel SVM implementations exist, but not their comparative study with any benchmark dataset.
- Mostly analyzed for binary classification.
- We use the MNIST hand written digit dataset to analyze performance and accuracy of three parallel algorithms.



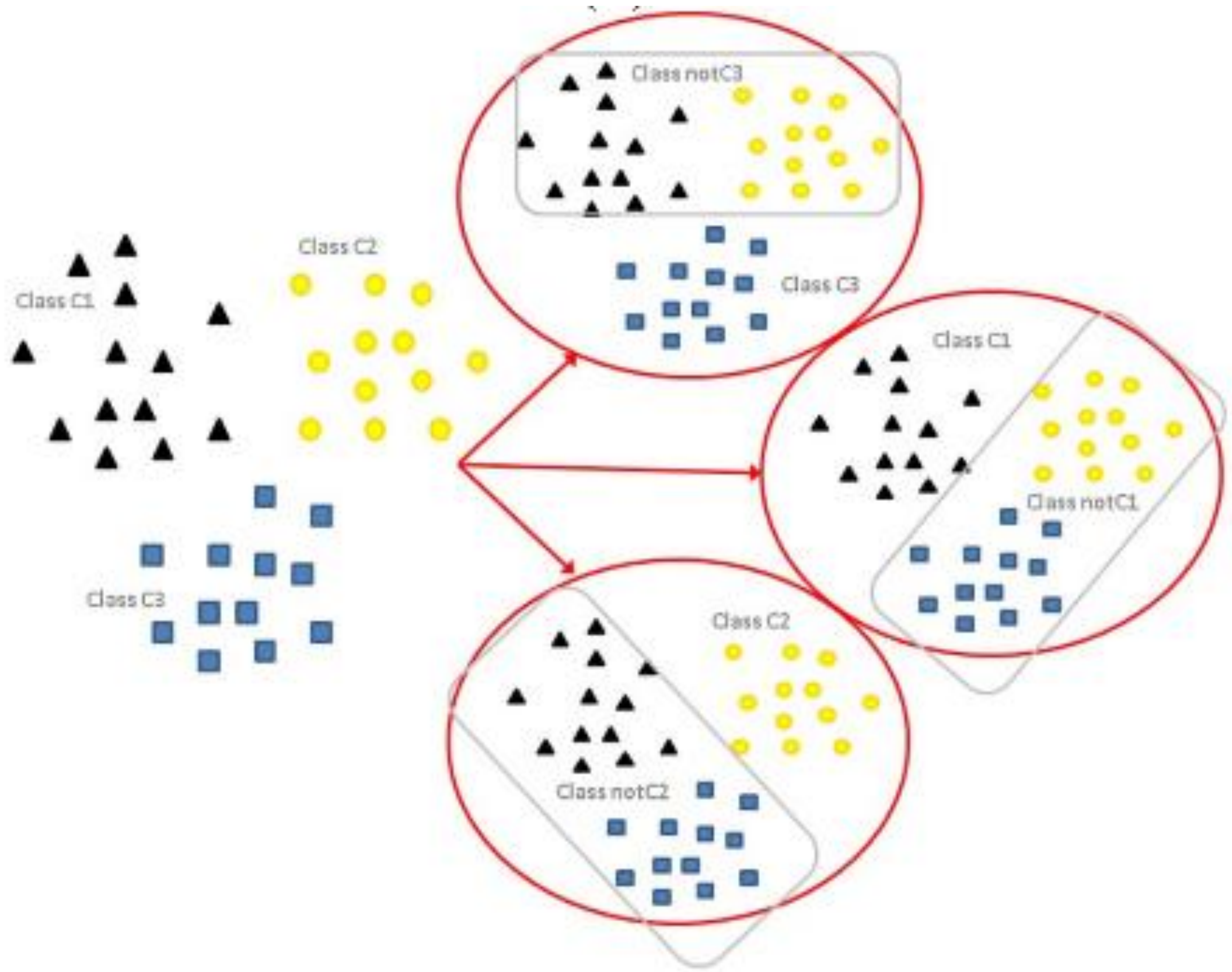
# Support Vector Machines



# Increased computation : Kernels

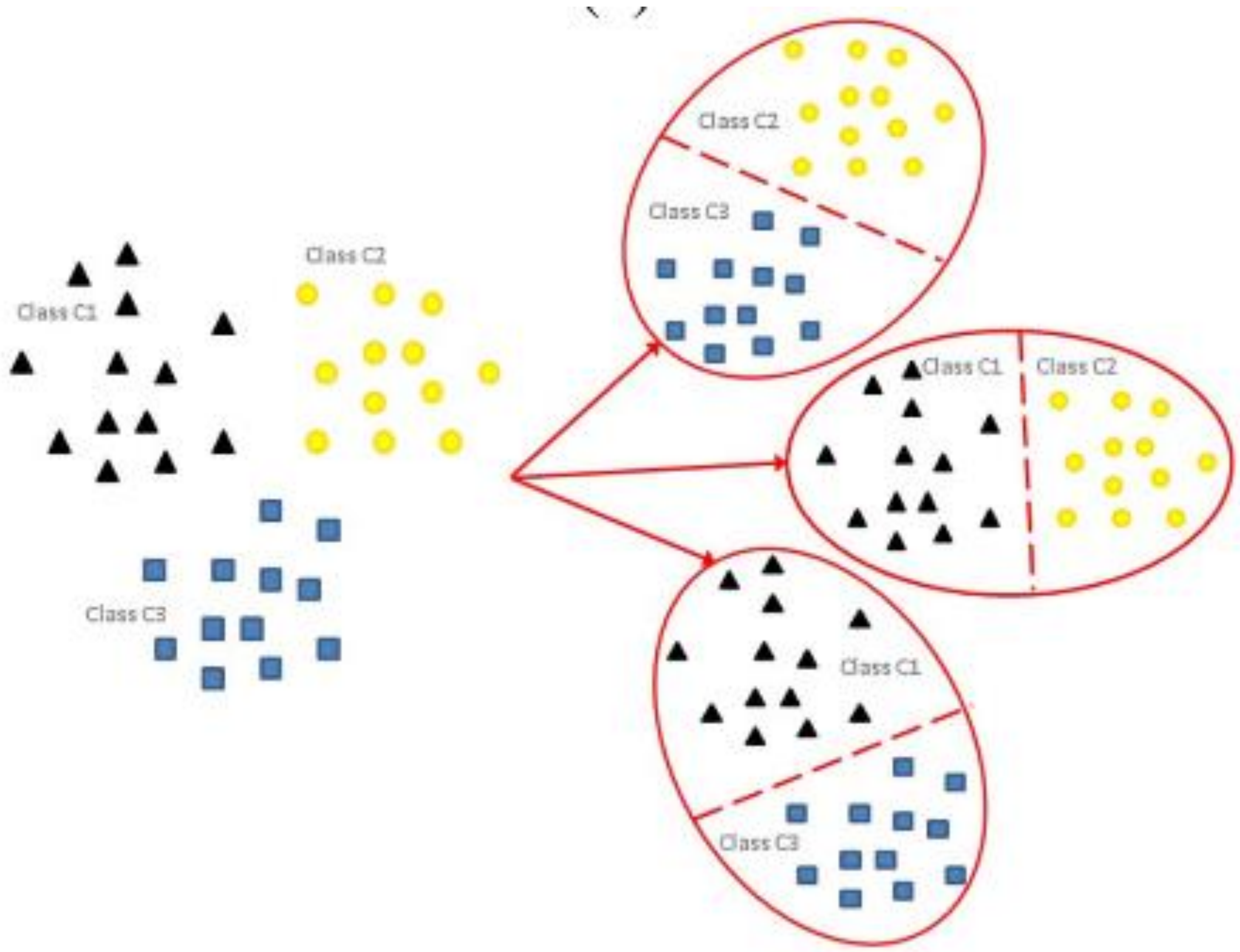


# Increased computation : Multi Class Classification




N Binary SVMs

# Increased computation : Multi Class Classification

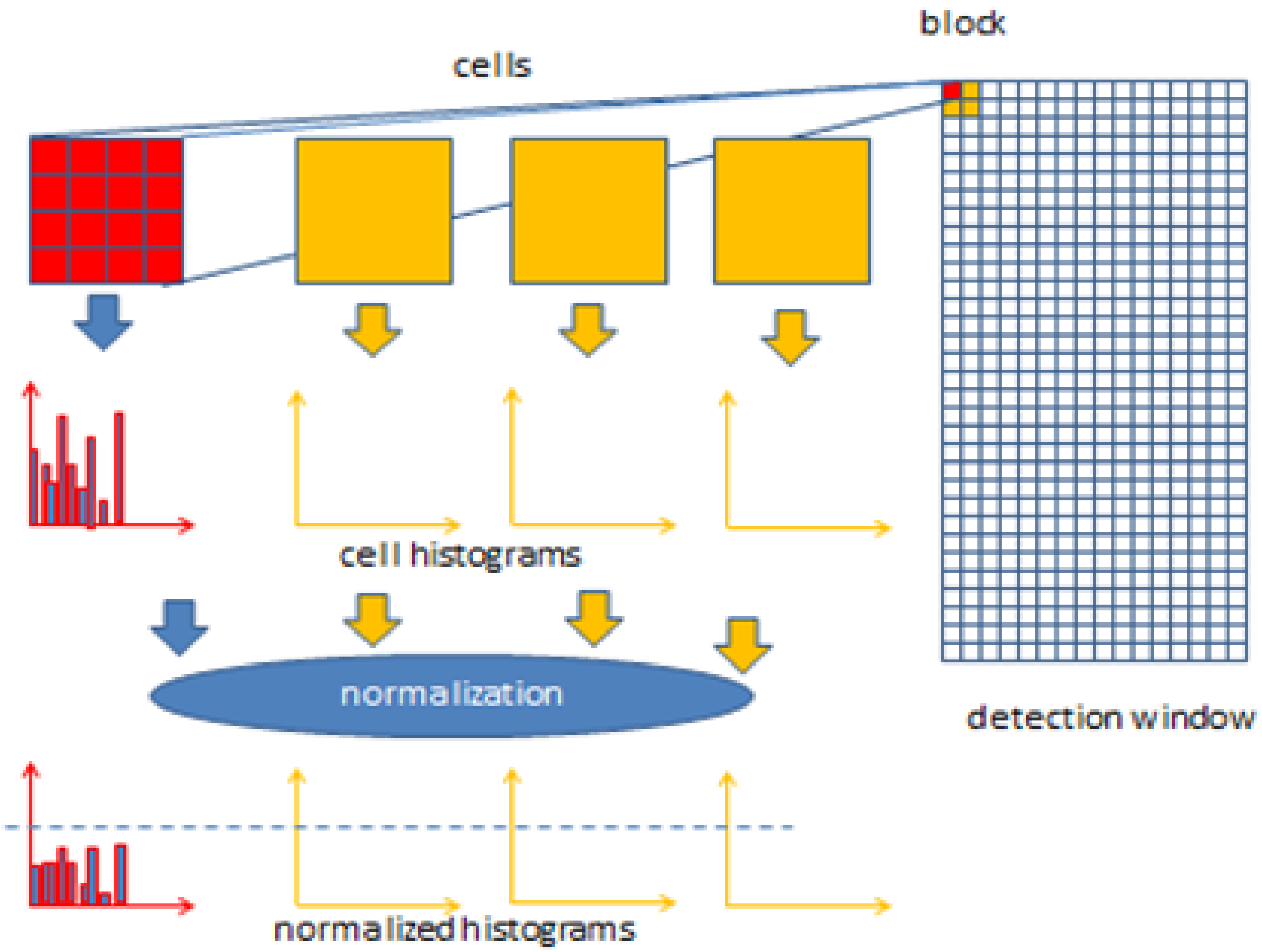



N*(N-1)/2 Binary SVMs

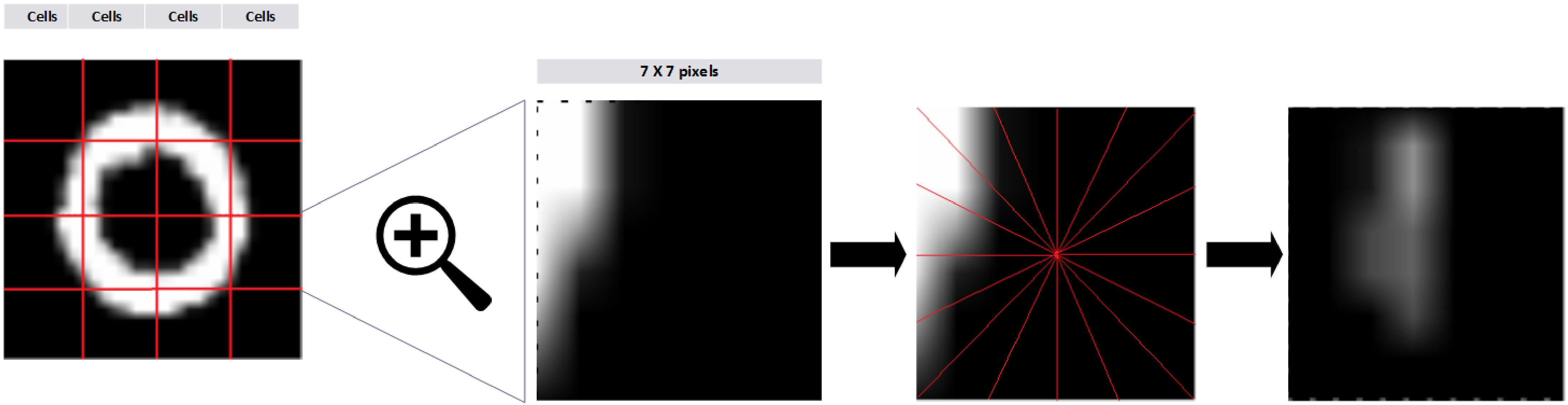
# The dataset : MNIST

- Mixed National Institute of Standards and Technology database
- 42,000 data points
- 28 X 28 pixels

# Data Preprocessing : Histogram of Oriented Gradients (HOGs)

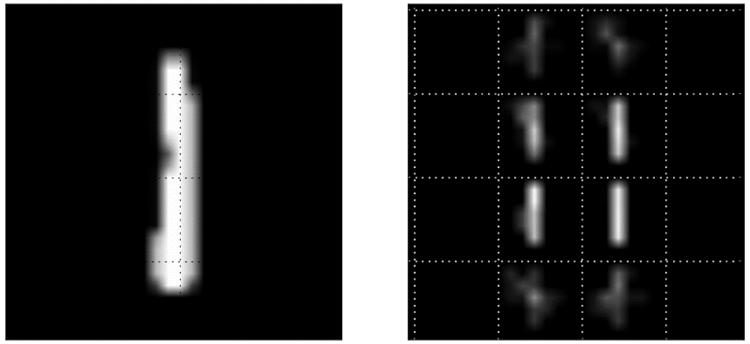


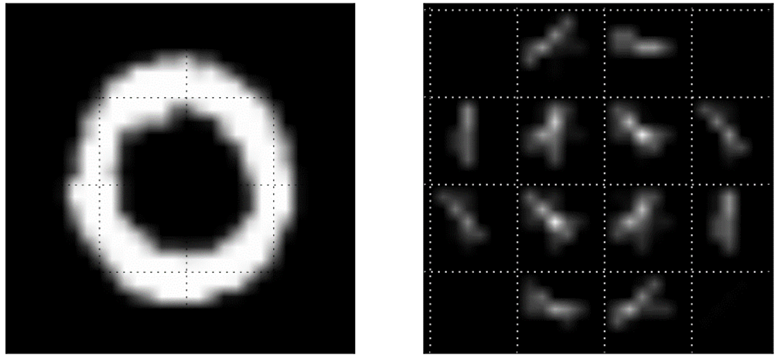
# Data Preprocessing : Histogram of oriented gradients

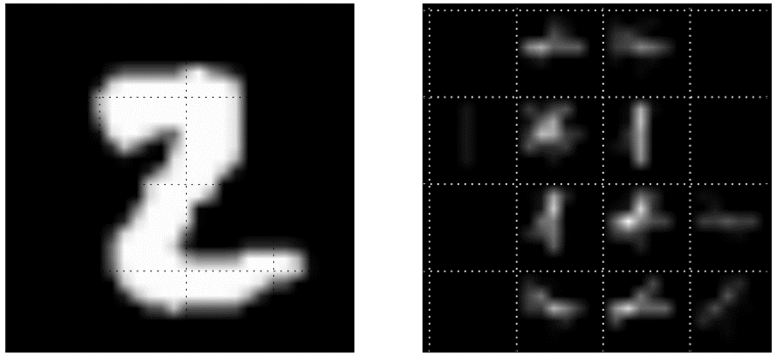


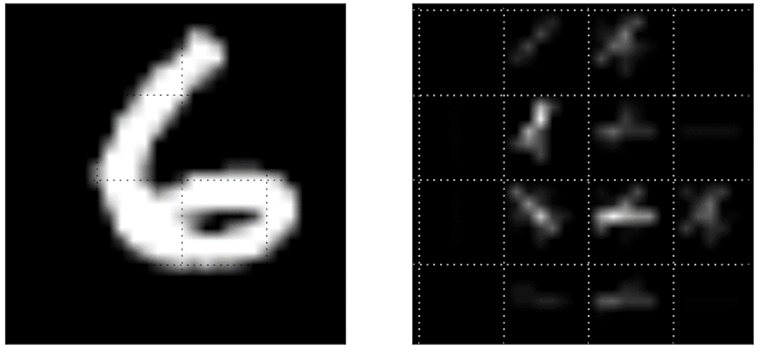

256 features

# Digits after preprocessing

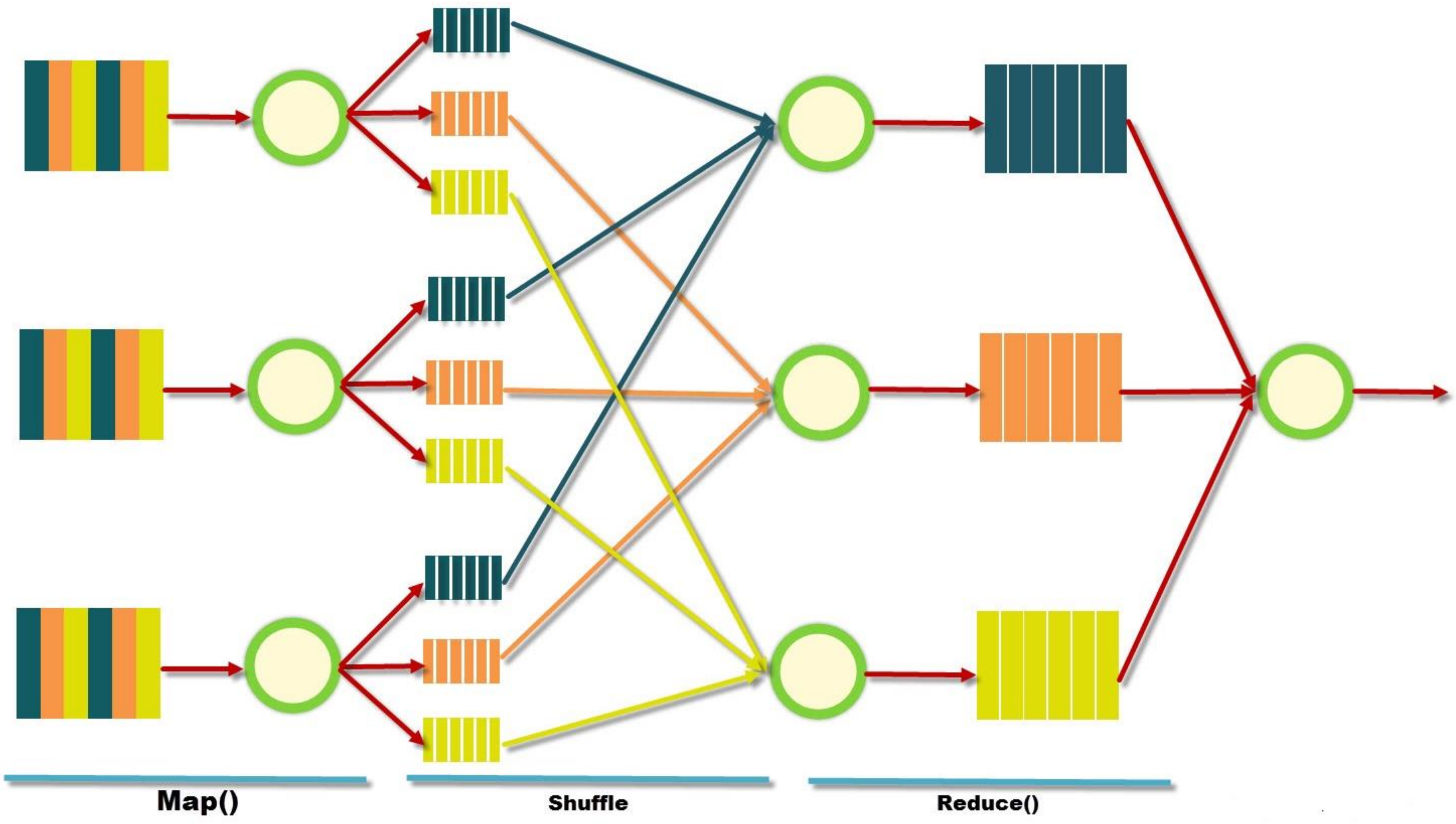




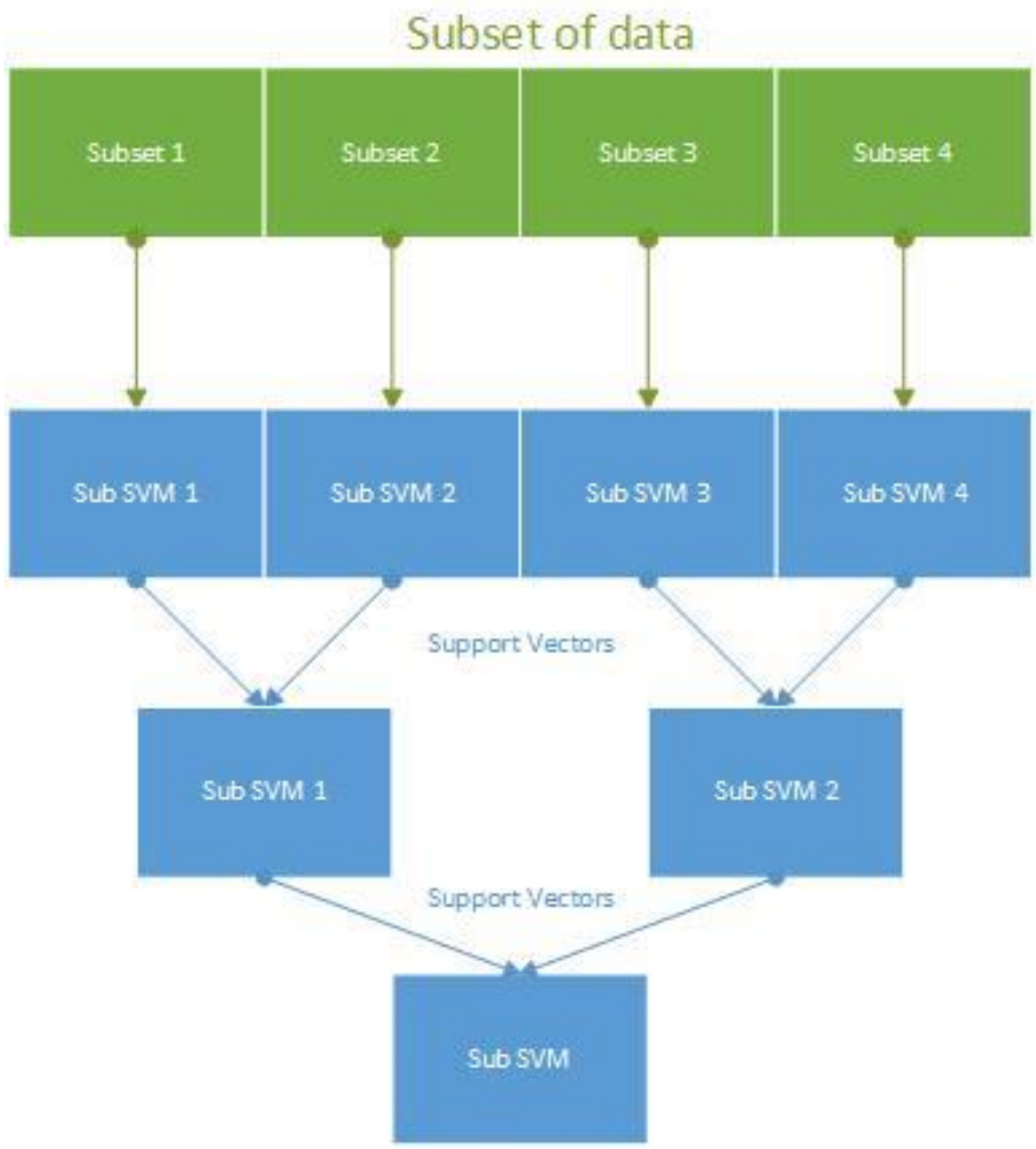




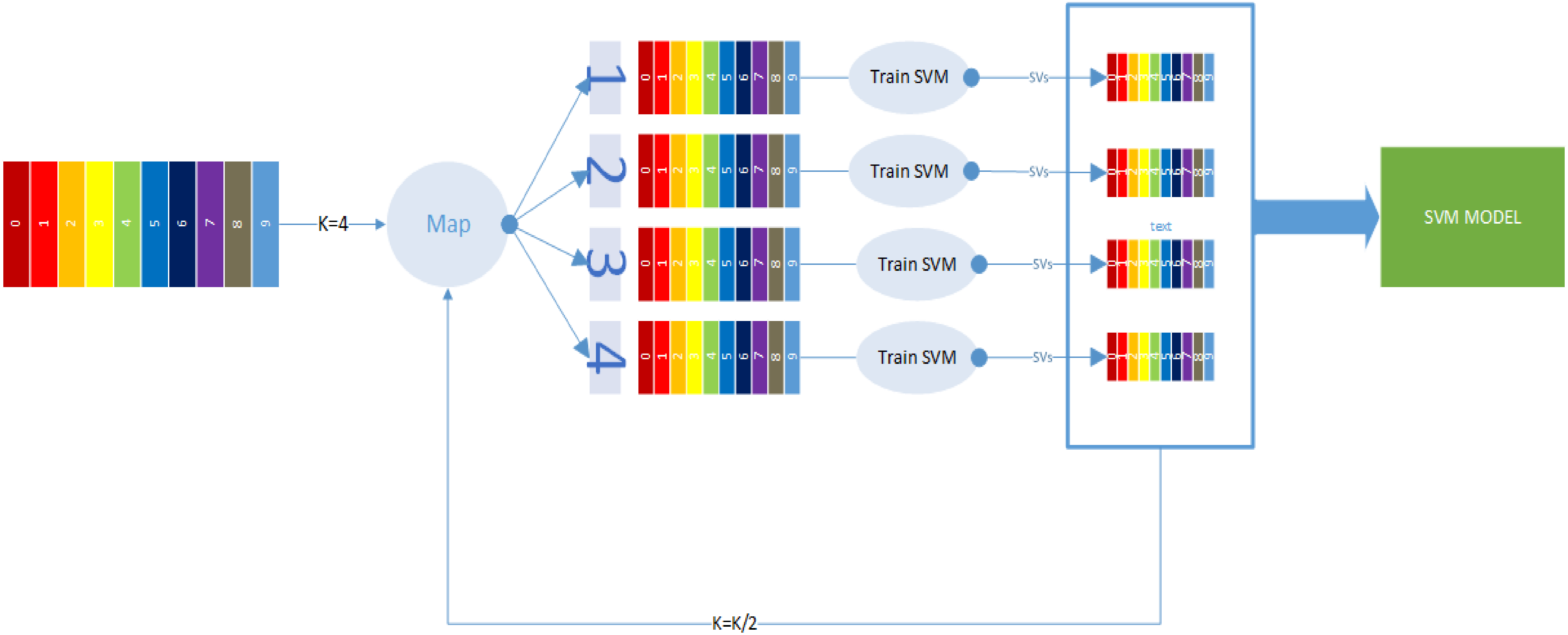
# MapReduce Model



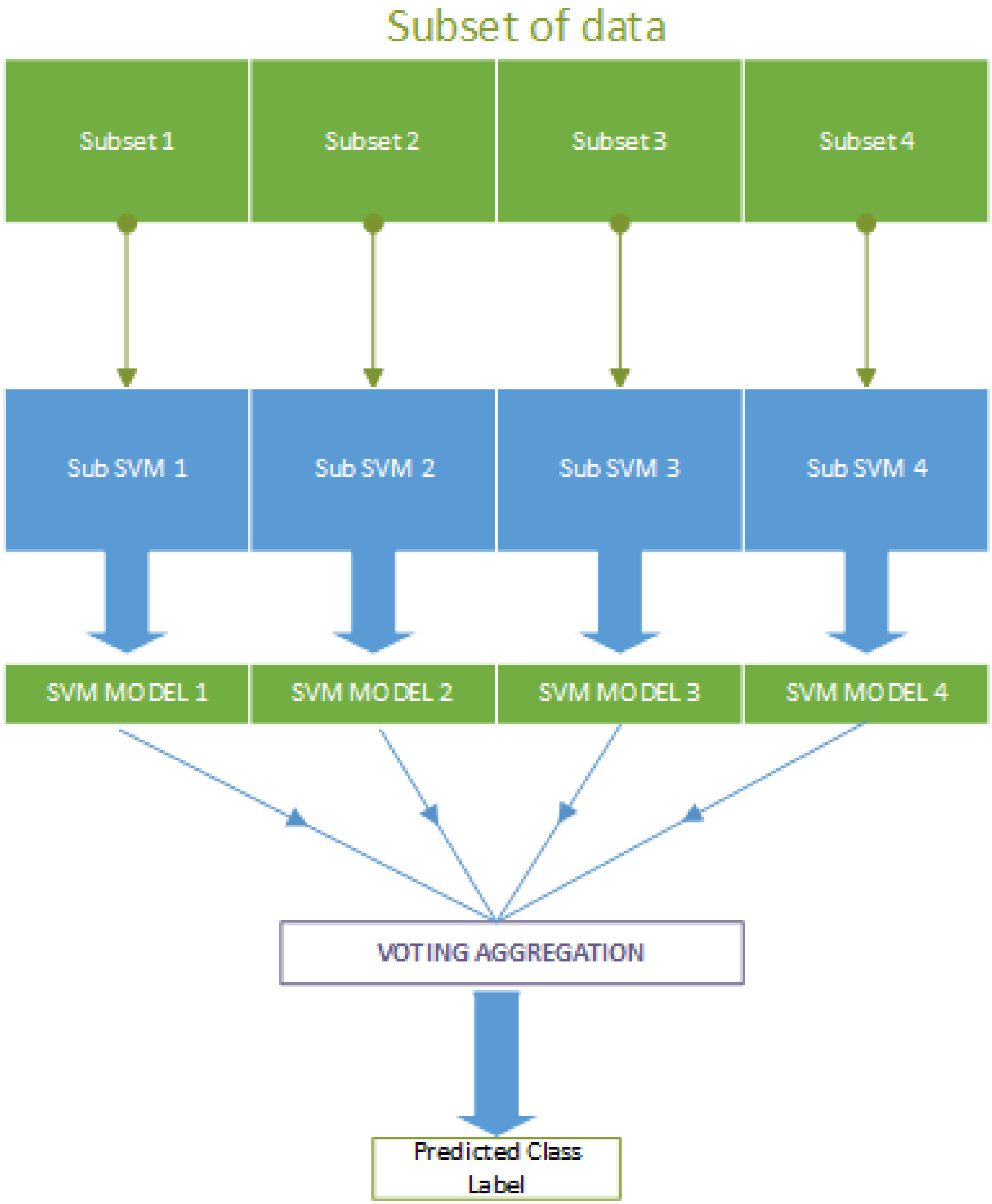
# Cascade SVM


Subset of data
Subset 1
Subset 2
Subset 3
Subset 4
Sub SVM 1
Sub SVM 2
Sub SVM 3
Sub SVM 4
Support Vectors
Sub SVM 1
Sub SVM 2
Support Vectors
Sub SVM

# Cascade SVM : MapReduce Implementation

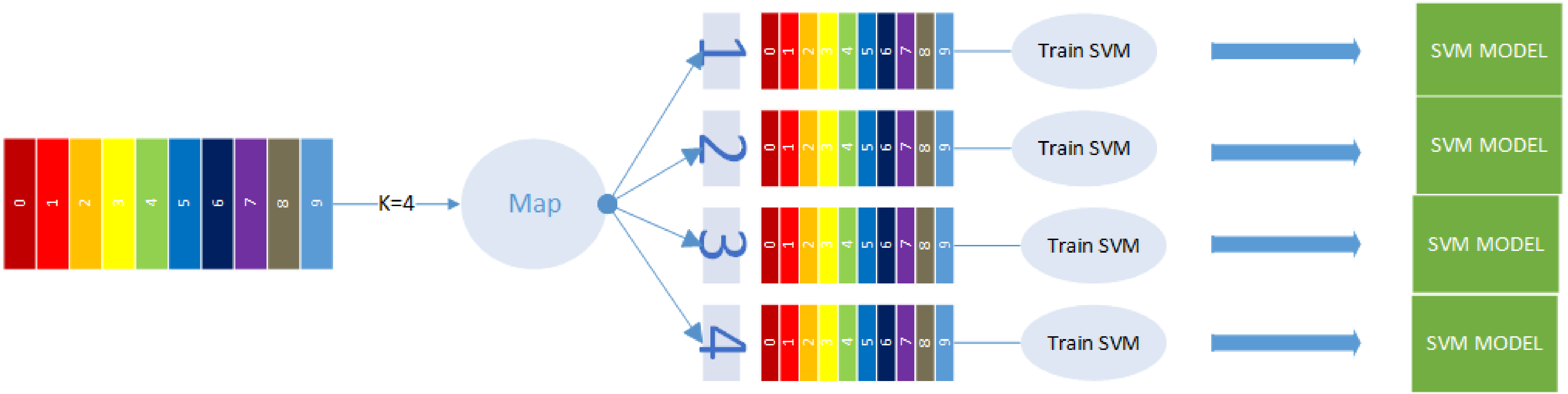


# Bagging SVM

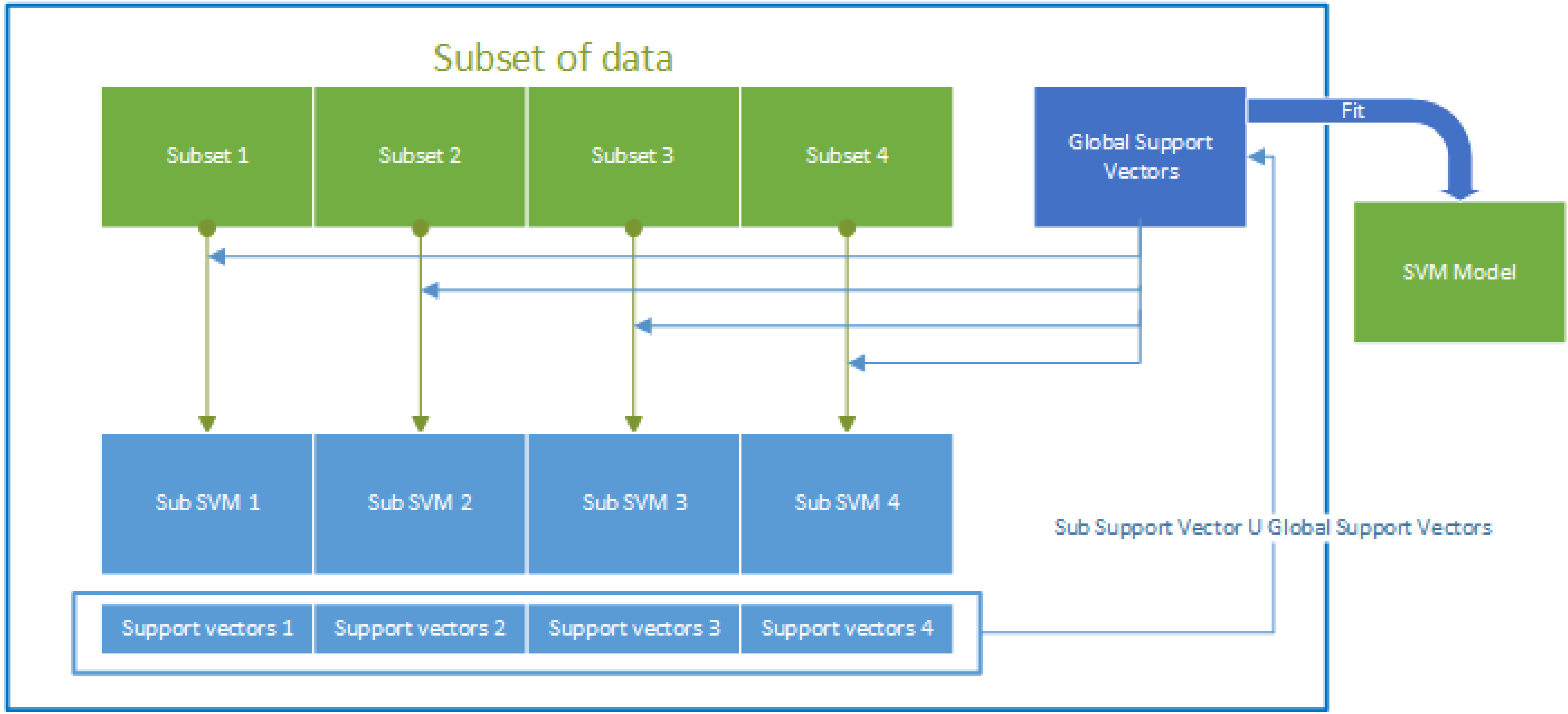


Subset of data
Subset 1
Subset 2
Subset 3
Subset 4
Sub SVM 1
Sub SVM 2
Sub SVM 3
Sub SVM 4
SVM MODEL 1
SVM MODEL 2
SVM MODEL 3
SVM MODEL 4
VOTING AGGREGATION
Predicted Class Label

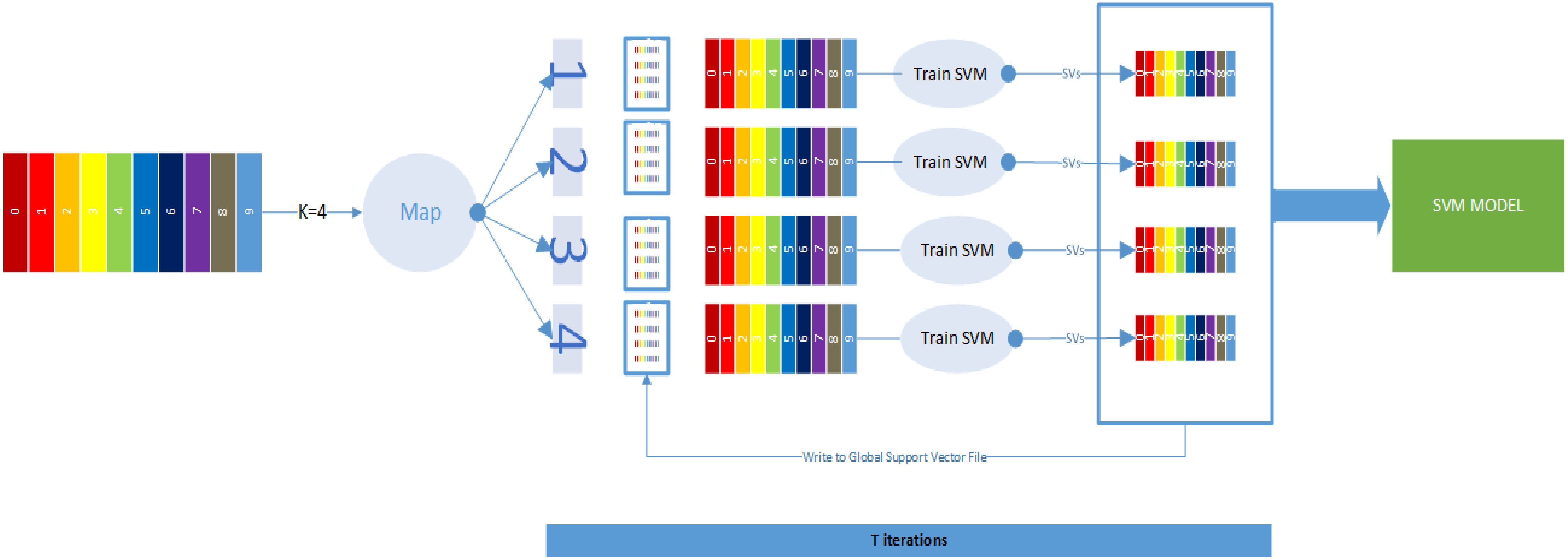
# Bagging SVM : MapReduce Implementation



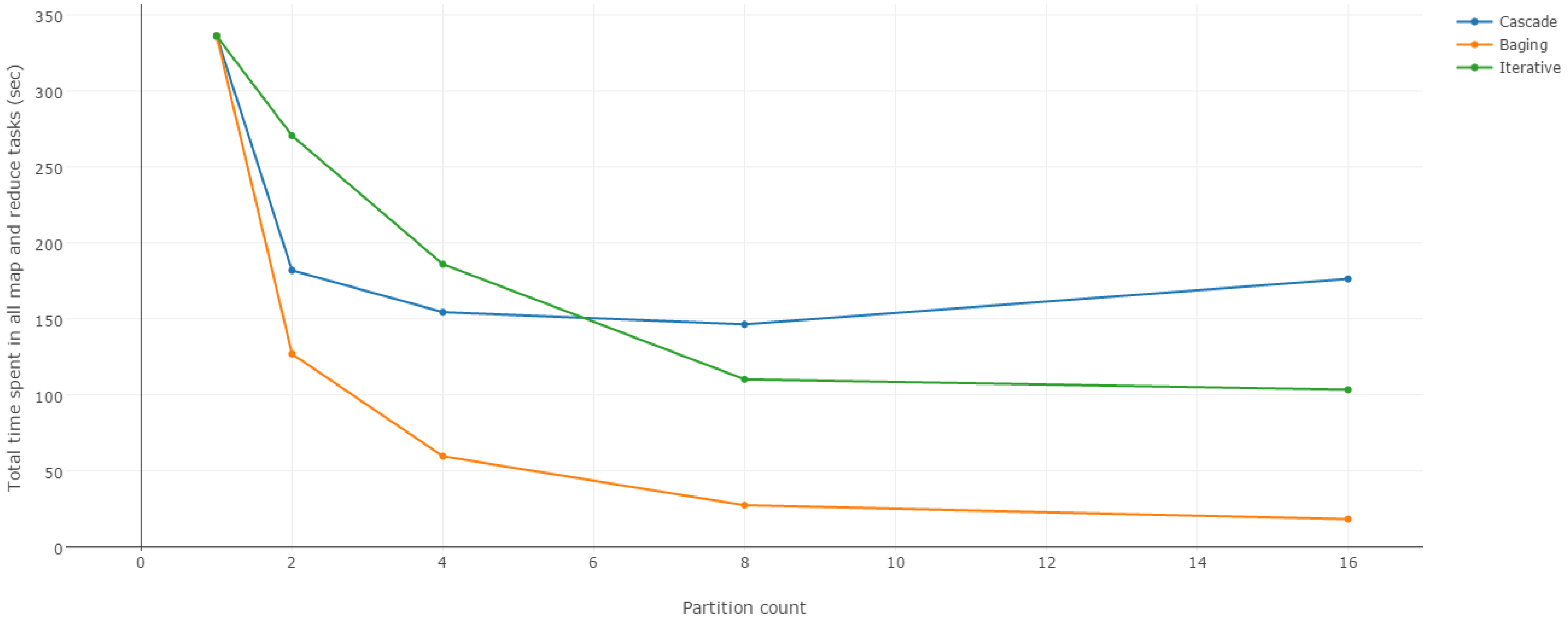
# Iterative SVM



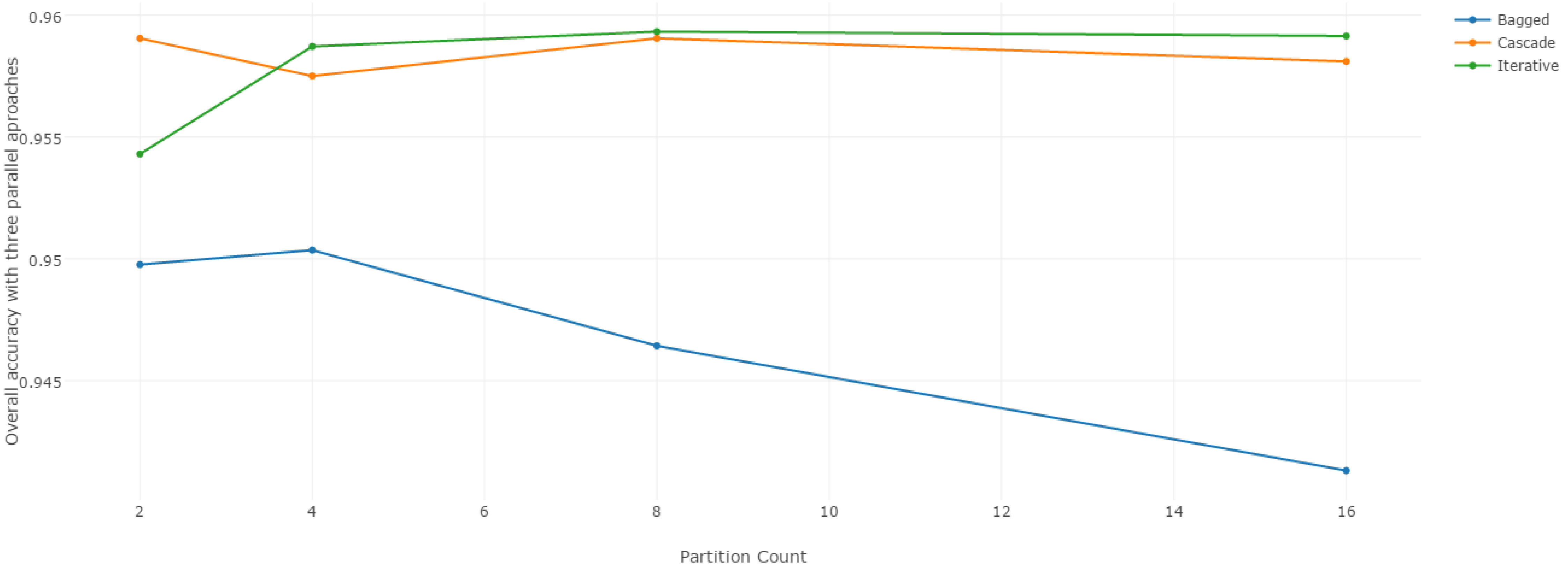
# Iterative SVM : MapReduce Implementation



# Training time vs Partition count



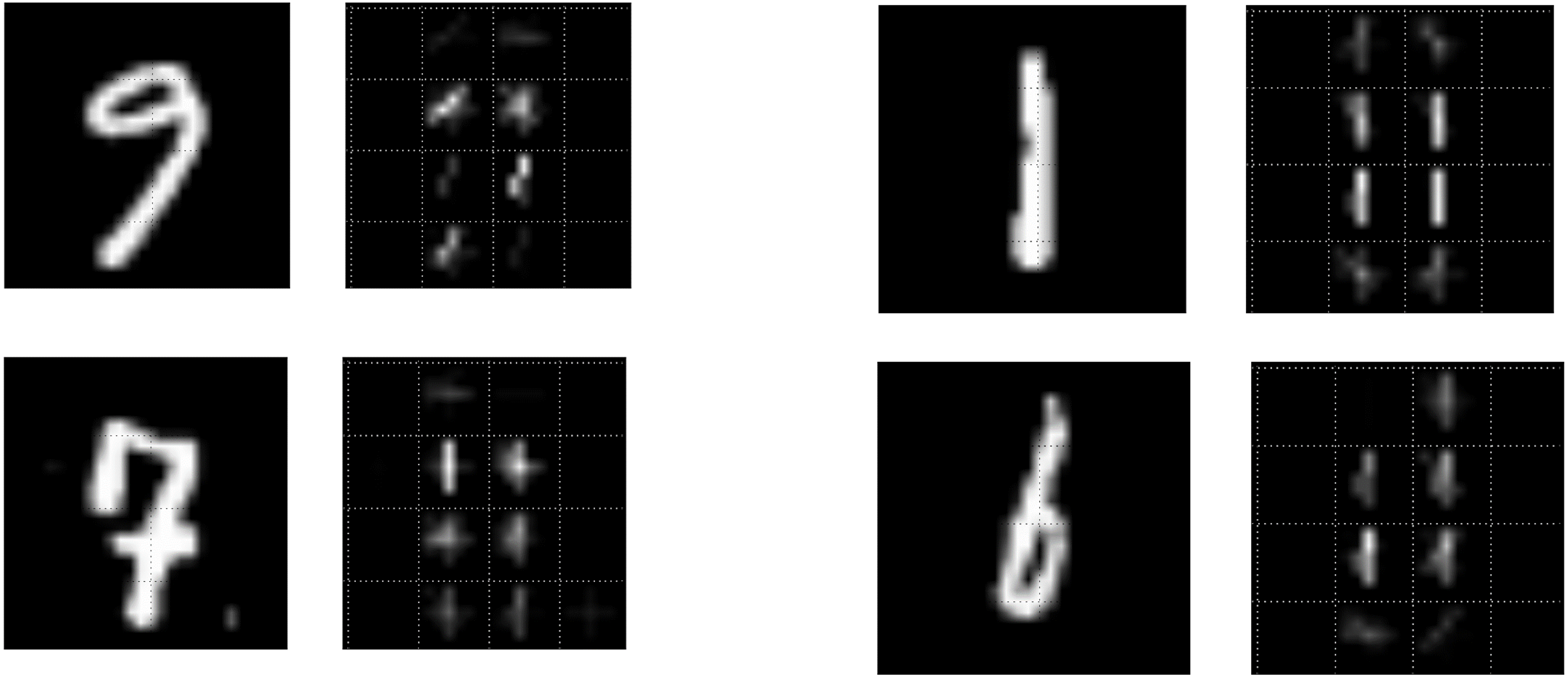
# Accuracy


0.96
0.955
0.95
0.945
Overall accuracy with three parallel aproaches
2
4
6
8
10
12
14
16
Partition Count
Bagged
Cascade
Iterative

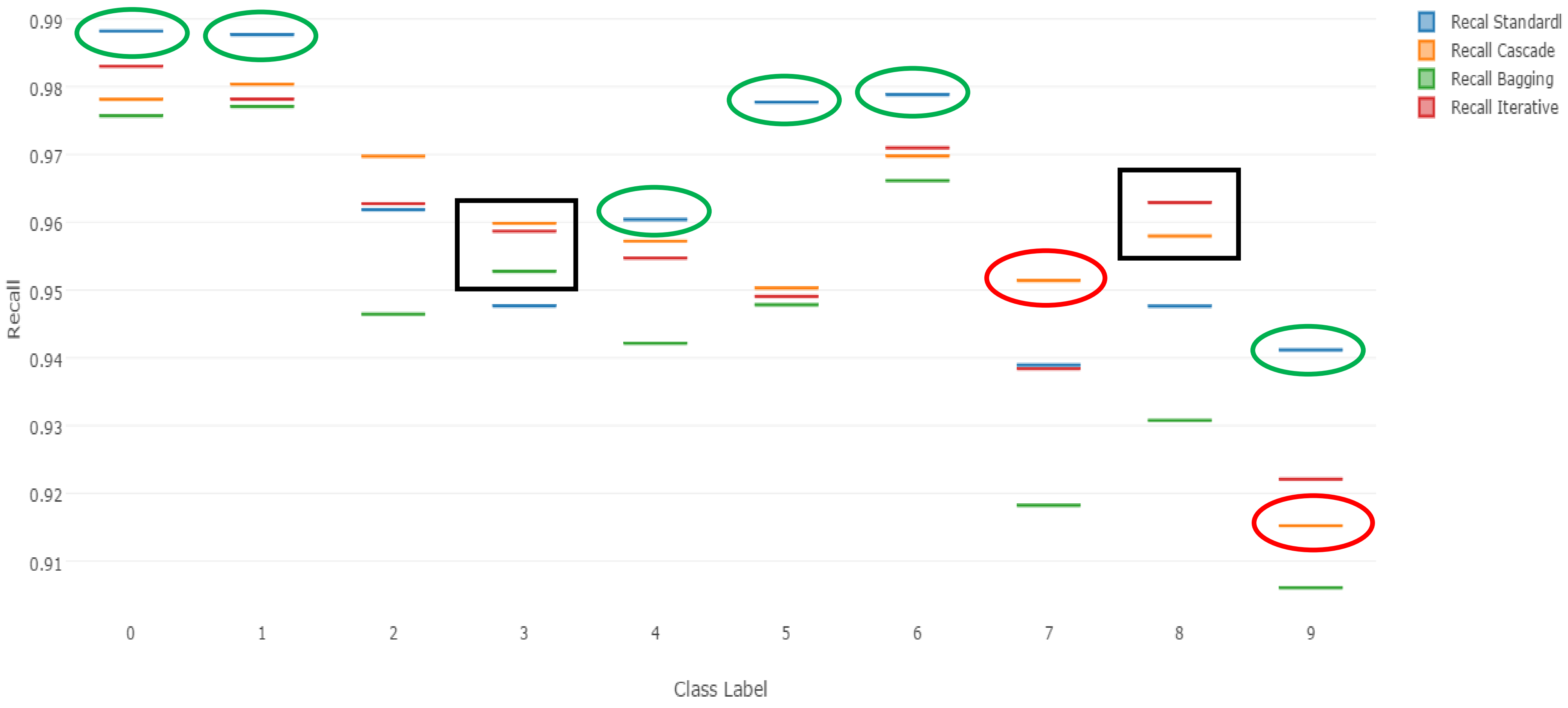
# Standard SVM confusion matrix

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 839 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 1 | 1 |
| 1 | 1 | 882 | 1 | 0 | 2 | 0 | 1 | 4 | 1 | 1 |
| 2 | 4 | 4 | 782 | 9 | 4 | 5 | 1 | 2 | 2 | 0 |
| 3 | 4 | 0 | 11 | 815 | 1 | 13 | 0 | 3 | 10 | 3 |
| 4 | 1 | 1 | 3 | 0 | 825 | 0 | 4 | 4 | 3 | 18 |
| 5 | 0 | 0 | 2 | 5 | 1 | 790 | 3 | 0 | 6 | 1 |
| 6 | 8 | 1 | 0 | 0 | 2 | 4 | 832 | 0 | 3 | 0 |
| 7 | 0 | 6 | 4 | 5 | 7 | 1 | 0 | 815 | 3 | 27 |
| 8 | 6 | 8 | 6 | 2 | 0 | 8 | 5 | 1 | 742 | 5 |
| 9 | 1 | 2 | 1 | 6 | 15 | 2 | 1 | 17 | 3 | 768 |

# False Negatives



# Recall comparison



# Support Vectors in the final model

| | |
|---|---|
| **Standard SVM** | **6340** |

| | Cascade | Bagging | Iterative |
|---|---|---|---|
| 2 | 6026 | 8137 | 7024 |
| 4 | 5941 | 10554 | 7748 |
| 8 | 5884 | 13909 | 8522 |
| 16 | 5888 | 18143 | 8921 |

# Conclusion

- The decrease in accuracy is only 0.5% to 3% for all parallel approaches, making all of them usable.
- Iterative SVM is the best choice for high partition counts.
- Bagging SVM takes lowest training time, with 3% reduction in accuracy. They can be used for initial approximation on massive datasets.
- Cascade SVM gives most relevant support vectors with high accuracy.

# Future work

- Performance of parallel SVM algorithms on unbalanced data
- Quantification of communication cost between mappers and reducers.
- More sophisticated method to calculate training error in Iterative SVM

# Acknowledgement

- Dr. Stan Thomas
- Dr. David John and Dr. William Turkett
- Dr. Todd Torgersen
- Department of Computer Science
- Friends and Family

# Precision comparison