Phishing Website and Letter Recognition Using Supervised Learning

Georgia Institute of Technology CS 7641: Machine Learning Assignment 1

Yan Cai

GT ID: ycai87

Abstract

This paper applies different supervised learning algorithms to analyze on two distinct datasets: phishing websites and letter recognition. In particular, decision tree, K-nearest Neighbors, boosting, support vector machines and neural networks are the learning algorithms applied. Performance for different models are also compared in the analysis. The whole analysis is done using Weka GUI.

Why Are The Problems Interesting?

Two problem sets are identifying the phishing websites and recognizing letters from the image. One the biggest threats to online security is phishing websites. The phishing websites usually has the layout as the legitimate websites and it is important for anti-malware firewall and internet browser to detect it, either from webpages or moreover, from the spam emails. Using machine learning techniques to classify the websites is of practical use. The other problem is to recognize the letter from the image. Image recognition is a broad topic, and letter recognition based on machine learning is an interesting application. Letters are able to be recognized by computer vision through the scanning of the text. Books can be electronically stored online after the whole contents are identified.

Both datasets are classification problems. They have more than 10K instances with more than 15+ features. It provides the opportunity to apply different machine learning techniques on these non-trivial datasets.

Datasets Introduction

Both datasets come from UCI machine learning repository. The phishing websites dataset has 11055 instances, 31 attributes and 2 classes. The two classes are binary values, indicating whether the instance is a phishing website. The letter recognition dataset has 20000 instances, 17 features and 26 classes. The 26 classes are 26 English letters.

Methodology

All the Each data set is 70-30 split into two datasets. 70% of the original dataset is training set. The remaining 30% dataset is test set. Various supervised learning algorithms are applied on the training set first and then on the test set. Hypeparameters are tuned through 10 fold cross validation on the training set in the model selection. The model is selected based on the best cross validation score, which is the classification accuracy on the training set. Test set will not be touched during the model selection process and will only be used for testing the model. Regarding the parameters for each model, the values are first evaluated using Weka default values, which are rule of thumb values usually. All models are biased towards simplicity and smoothness. A detailed analysis and the learning curve will be given in each of the learning algorithms. Each learning model selected for learning curve is highlighted in the parameter table.

Decision Tree

For decision tree, J48 is used. The following parameters are tuned:

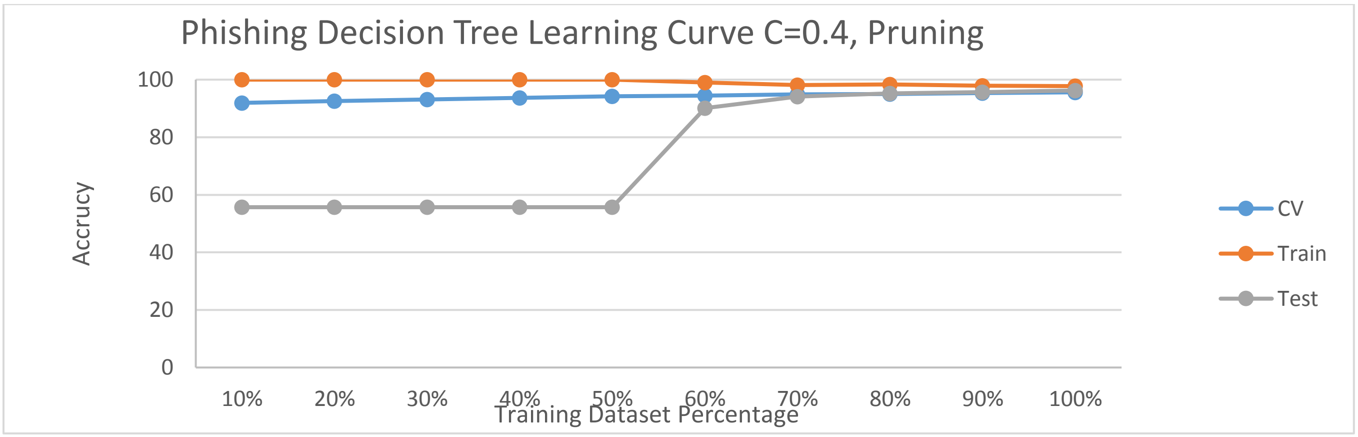
- Confidence factor. The confidence factor sets the confidence threshold for pruning. Default is 0.25.
- Unpruned or pruned. It indicates if pruning is performed.

Pruning is often used to reduce the size of tree. However, pruning would decrease the accuracy of the training set, but in general increases the accuracy for test data. It is used to mitigate overfitting, in case the model achieves too good accuracy for training data, but at the same time too specific and performs poorly on test data. The training is usually fast. The minimal instance at leaf node is set 2 at default.

Phishing Website

The parameter table and learning curve:

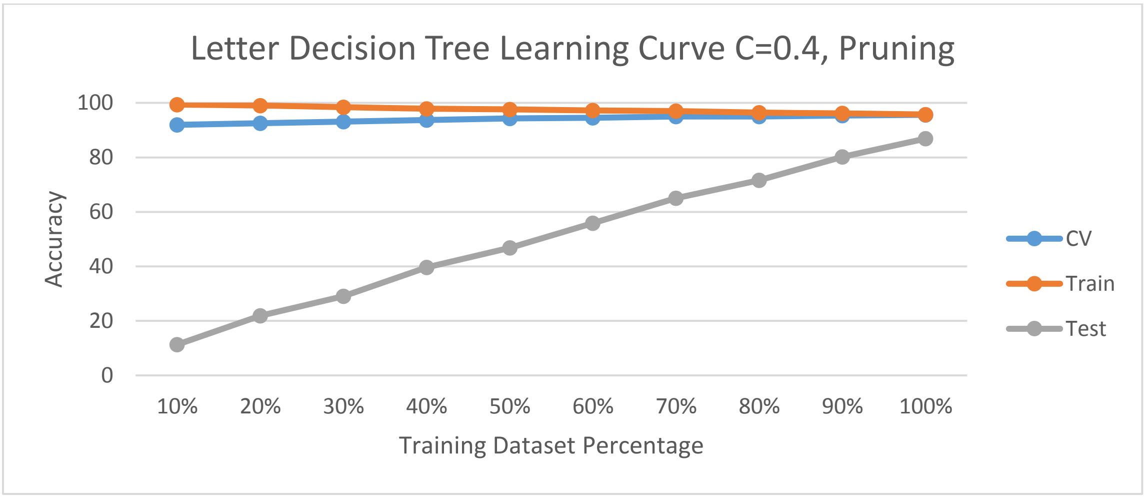
Confidence Factor	Pruning	Minimum Instance	Training Accuracy	CV-10 Accuracy	Testing Accuracy
N/A	Yes	2	98.3198	95.4753	96.2327
0.1	No	2	96.7688	94.817	96.1423
0.25	No	2	97.5055	95.231	96.3231
0.4	No	2	97.7511	95.605	96.2327
0.6	No	2	98.3327	95.45	96.2025
0.8	No	2	98.3327	95.45	96.2025



Letter Recognition

The parameter table and learning curve:

Confidence Factor	Pruning	Minimum Instance	Training Accuracy	CV-10 Accuracy	Testing Accuracy
N/A	Yes	2	95.9826	85.267	86.6744
0.1	No	2	95.0247	85.2384	86.6079
0.25	No	2	95.5679	85.3528	86.8242
0.4	No	2	95.7324	85.37418	86.8574
0.6	No	2	95.9826	85.31698	86.8408
0.8	No	2	95.9826	85.31698	86.8408



As we can see for phishing data table, pruning decreases the accuracy, but the overall CV score is close, indicating the input variance in different datasets impacts little. When confidence facto is 0.4, it achieves the best CV score. The testing results also indicate that C = 0.4 is the best model. As we can see from the letter recognition table, performance improves when pruning is selected. However, the negligible difference in score indicates that the model relies relatively little on the input variations. The confidence factor equal to 0.4 best fits the model. The little difference regarding pruning observed in letter recognition set implies that the dataset is not easily separable.

In terms of running time, the training takes a lot time than testing, indicating it is an eager learner. The CV and Training converges to high value indicating more training data helps achieves better performance.

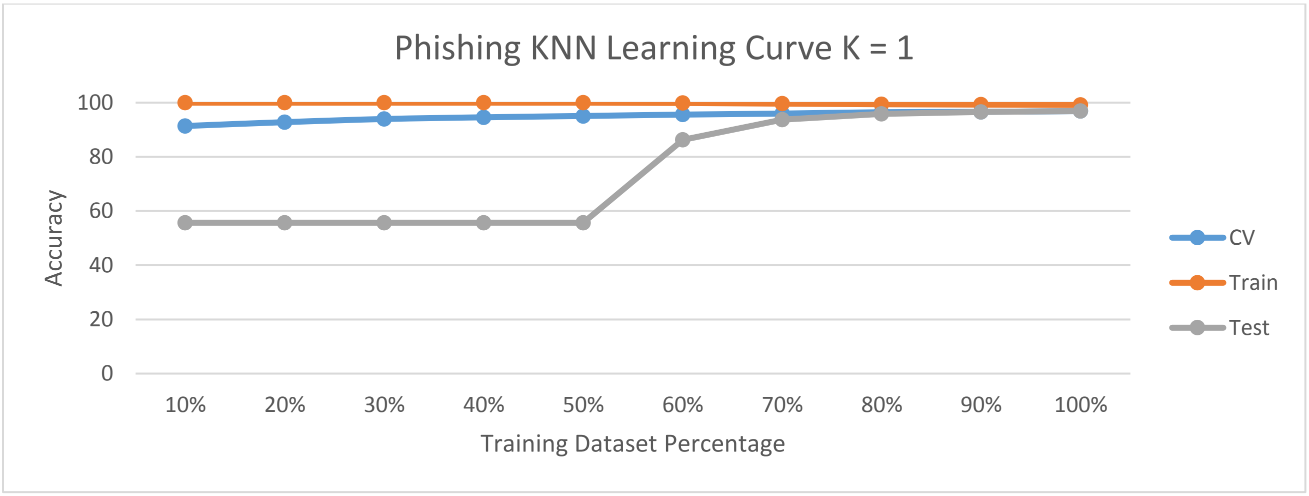
K-Nearest Neighbors

For KNN, IBK learner is used in Weka. Parameter K, the number of neighbor to use is tuned. There is no distance weight and Euclidean distance is used in the algorithm.

Phishing Website

The parameter table and learning curve:

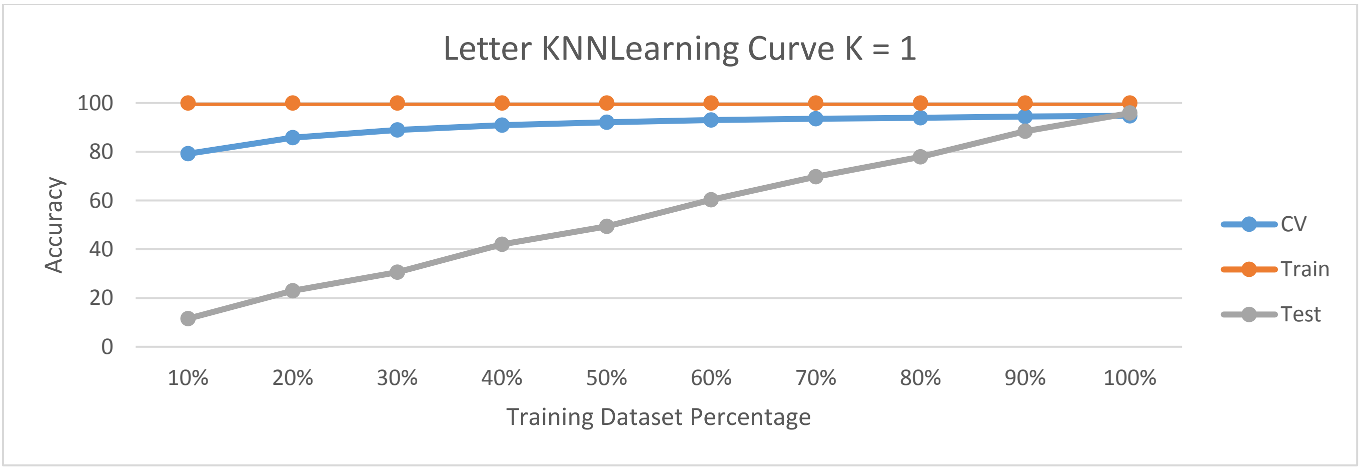
K	Training Accuracy	CV-10 Accuracy	Testing Accuracy
1	99.134	96.88498451	96.8957
5	96.2393	94.77817223	95.2984
10	95.0885	94.28701559	94.4545
30	93.6409	93.33050867	93.5503
80	92.8008	92.71020321	92.2242



Letter Recognition

The parameter table and learning curve:

K	Training Accuracy	CV-10 Accuracy	Testing Accuracy
1	100	94.76009	95.841
5	96.9047	93.7808	95.0923
10	95.618	92.93728	94.3271
30	92.0008	89.57036	90.4342
80	84.7237	81.87141	83.1476



As we can see from the parameter table for phishing website, varying K does not make a big difference in CV score while in the letter recognition dataset, large K means significant performance decrease. It is also noted that K = 1 achieves the best performance for both datasets. The above observation indicates that for phishing website, the dataset has high repetition of instances that have similar features. For letter recognition dataset, poorer performance at higher K indicates that the dataset feature is more equally weighted and increasing K would significantly increases bias and decrease performance. KNN is a lazy learner and the test time is corresponding to the number of instances. The learning curves show that CV and training score converges to a high value, indicating this is a good model and more training dataset helps.

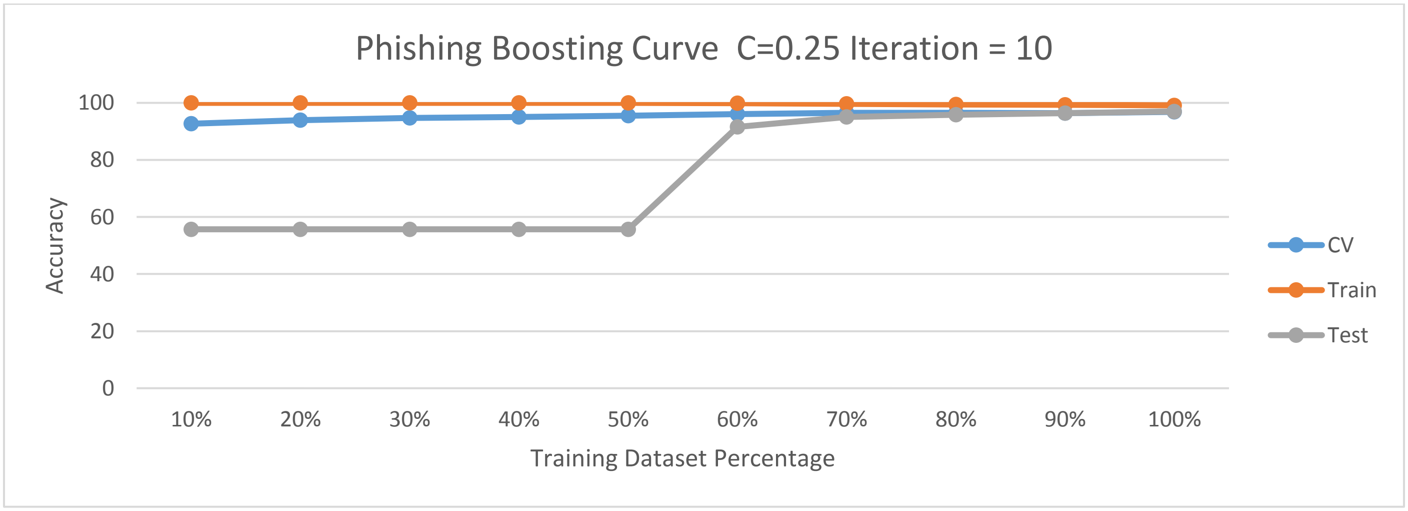
Boosting

Adaboost algorithm is used in boosting with decision tree as underlying leaner. That is because typically decision tree is a weak learner and adaboost can significantly boost the decision tree performance through iterations. Decision tree confidence factor and boosting iteration number are tuned to compare model.

Phishing Website

The parameter table and learning curve:

Learner	Iterations	Training Accuracy	CV-10 Accuracy	Testing Accuracy
Decision Tree –C 0.25	10	99.134	96.8463	96.9861
Decision Tree –C 0.4	10	99.134	96.6524	97.0464
Decision Tree –C 0.25	20	99.134	96.8334	96.9861
Decision Tree –C 0.4	20	99.134	96.6524	97.1368

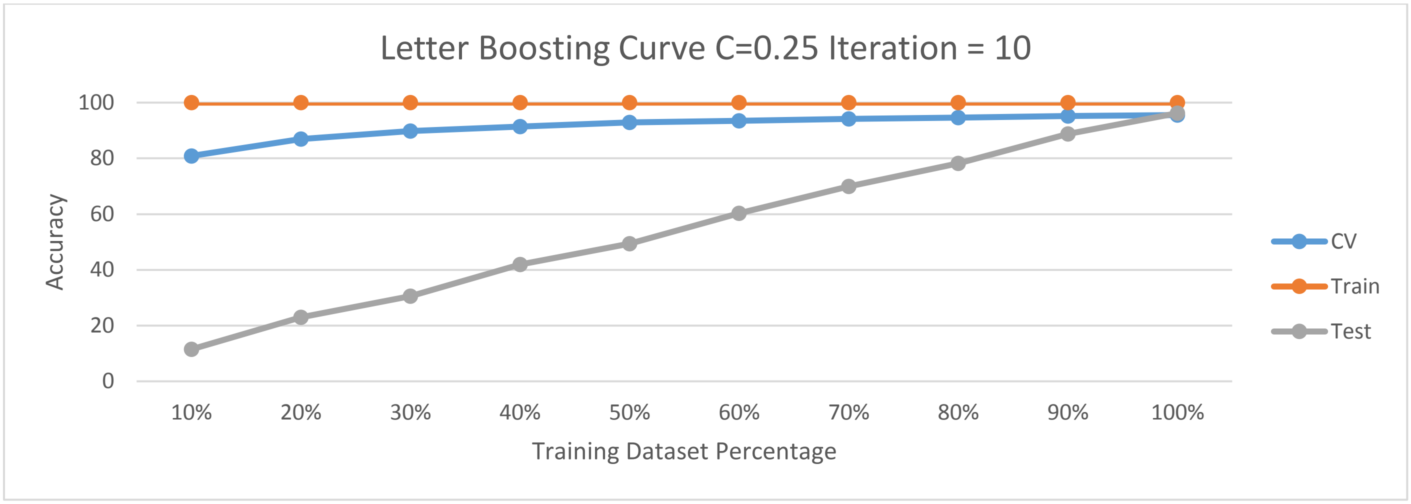


Letter Recognition

The parameter table and learning curve:

Learner	Iterations	Training Accuracy	CV-10 Accuracy	Testing Accuracy

Decision Tree –C 0.25	10	100	94.1668	94.7762
Decision Tree –C 0.4	10	100	94.2026	94.8262
Decision Tree –C 0.25	20	100	95.5168	95.9907
Decision Tree –C 0.4	20	100	95.5394	96.2236



As we can see from the table, Adaboost with J48 decision tree improves the performance compared with only J48 decision tree in previous example. For letter recognition dataset, the performance increases dramatically from ~86% to ~96%. This indicates that the decision tree model generalizes the dataset very well, and boosting effectively modified the weight within the underlying classifier. However, for the phishing website data, different confidence factor and iteration only makes little improvement in test score. This could be due to possibility that the dataset has a lot of noise instance since Adaboost is prone to noise. Since the previous decision tree model already fits the data very well, using Adaboost will not significant improve performance on validation/test data and could risk overfitting the data. That is why cross validation is needed.

The training time is correlated to the iterations and underlying learner. It is true that using a more iteration Adaboost model may improve performance, but the total training time with large number of iterations may be computationally preventive.

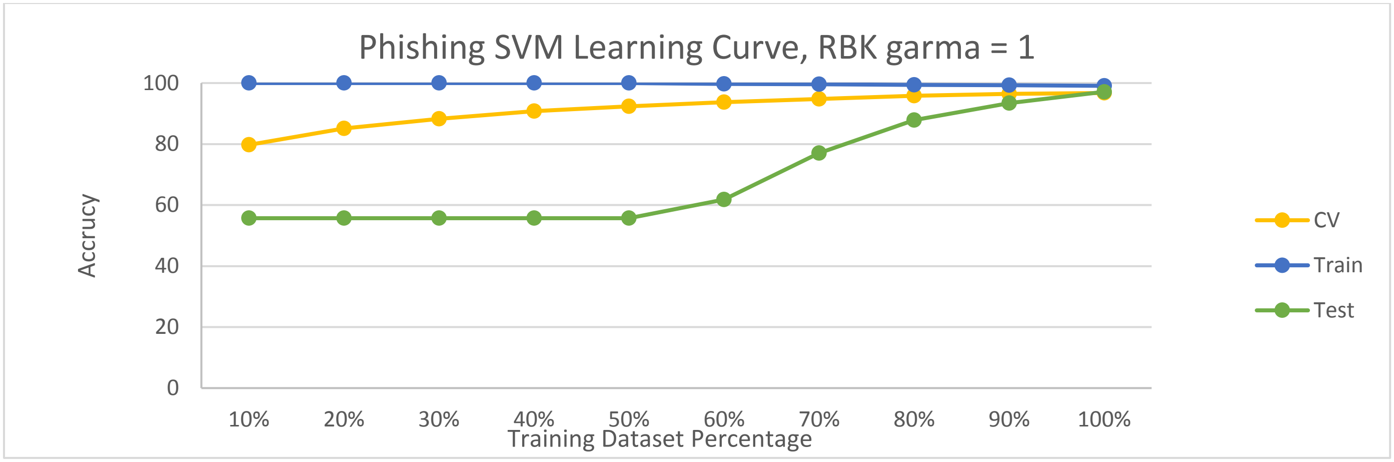
Support Vector Machine

Support Vector Machine or SVM is an eager learner. LibSVM classifier in Weka is used to evaluate the datasets. It creates a hyperplane and tries to separate the classes. It tries to optimize the distance between two instances and hyperplane. The shape of the hyperplane is determined by the kernel. In this experiment, we tried different kernels: linear kernel, polynomial with degree equal to 3, and radial basis function (RBK) with gamma equal to 1. The SVM type is C-SVM and cost is at default value 1

<u>Phishing Website</u>

The parameter table and learning curve:

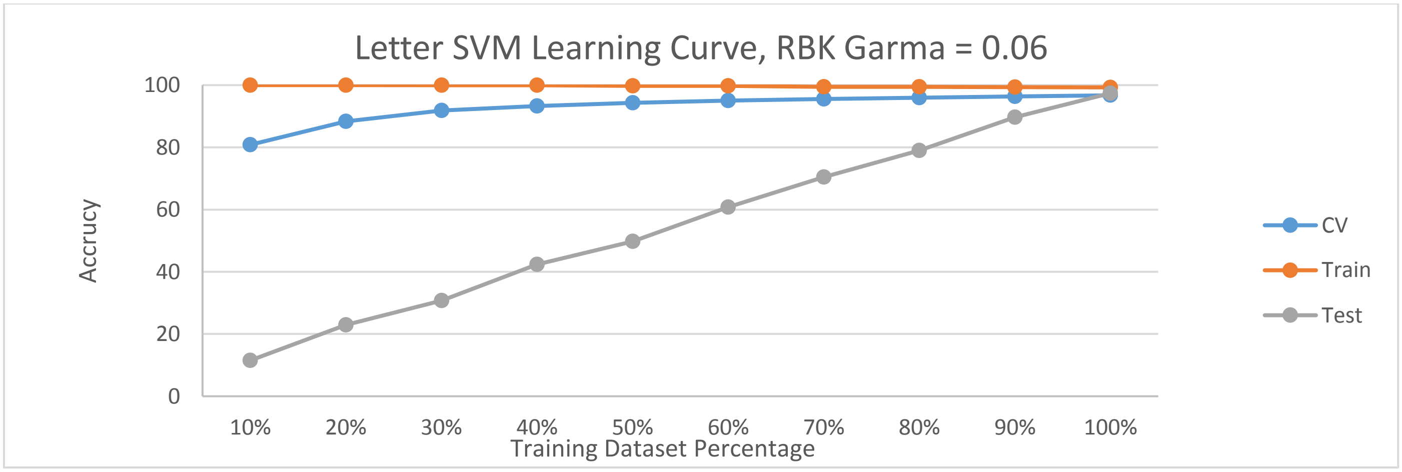
Kernel	Training Accuracy	CV-10 Accuracy	Testing Accuracy
Linear Kernel	94.0158	93.7314	93.8216
Polynomial garma = 0.03 degree = 3	94.3647	93.9253	94.5449
RBK garma = 1	99.0953	96.72996	97.1368



Letter Recognition

The parameter table and learning curve:

Kernel	Training Accuracy	CV-10 Accuracy	Testing Accuracy
Linear Kernel	87.5974	85.3528	85.094
Polynomial garma = 0.06 degree = 3	100	94.1597	94.7762
RBK garma = 0.06	99.2637	96.7189	97.4048



From the table for both datasets, RBK as the kernel choice performs the best. It also performs significantly better than linear kernel and polynomial kernel. For letter recognition dataset, polynomial kernel performs significantly better than linear kernel. The learning curve for both datasets suggest that more training data helps achieving better performance. It is also noted higher degree for polynomial kernel and bigger gamma value for RBK would significantly increases training time.

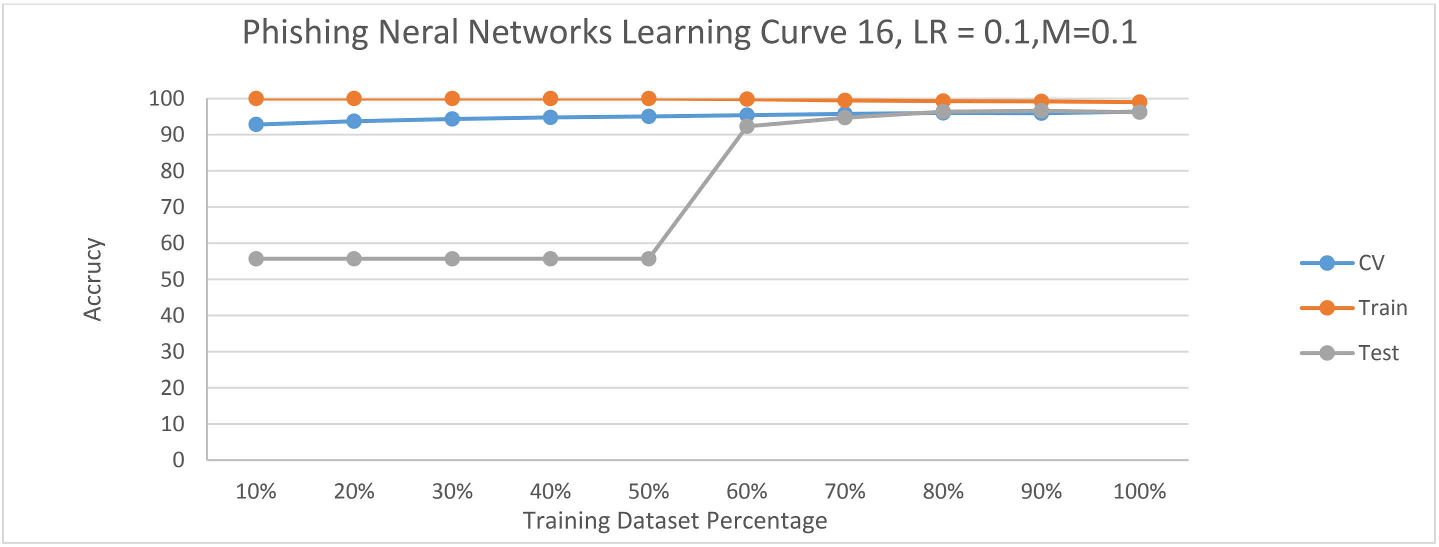
Neural Network

MultilayerPerceptron classifier in Weka is used to evaluate the datasets. The hidden layer is using the default setup: one layer with perceptron units = (attributes + classes) / 2. Thus for phishing website, the number is 16 and for letter recognition, the number is set as 20. We tuned the learning rate and momentum values in the experiment. Neural Networks is an eager learner, it uses back propagation to set the weight of different perceptions.

Phishing Website

The parameter table and learning curve:

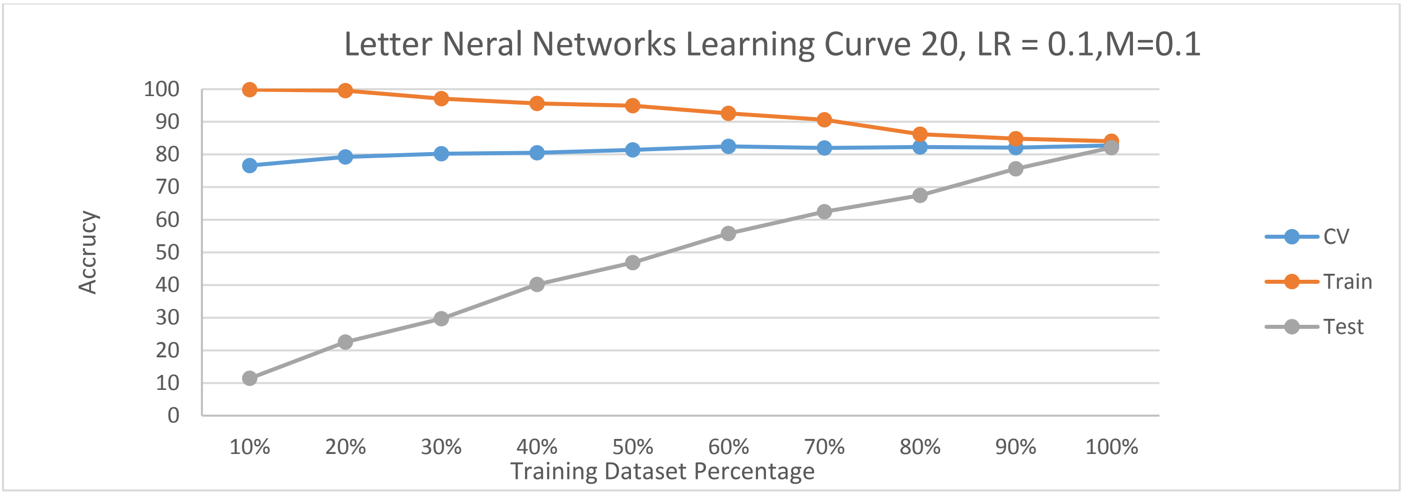
Layer	Training Accuracy	CV-10 Accuracy	Testing Accuracy
16, LR = 0.3 M=0.2	98.966	96.2518	96.4738
16, LR = 0.1,M=0.1	99.0048	96.381	96.1724
16, LR = 0.5,M=0.5	98.5912	95.9804	96.3864



Letter Recognition

The parameter table and learning curve:

Layer	Training Accuracy	CV-10 Accuracy	Testing Accuracy
20, LR = 0.3 M=0.2	83.7873	81.7499	81.9644
20, LR = 0.1,M=0.1	84.0946	82.765	82.1161
20, LR = 0.5,M=0.5	81.85	80.3488	78.5227



As we can see from the table, changing learning rate and momentum value does not significantly improve performance. Learning rate equal to 0.1 and momentum equal to 0.1 have the best performance for both datasets. Adding layers rather than using the default value is also tried and the performance difference is negligible. It also risks overfitting the data. This is probably because the current model already fits the data very well. However, increasing layers would dramatically increase the training time, thus it is not very much cost effective to run many layers of neural networks for both datasets. The Learning curves suggest that more data helps achieving better performance.

Summary

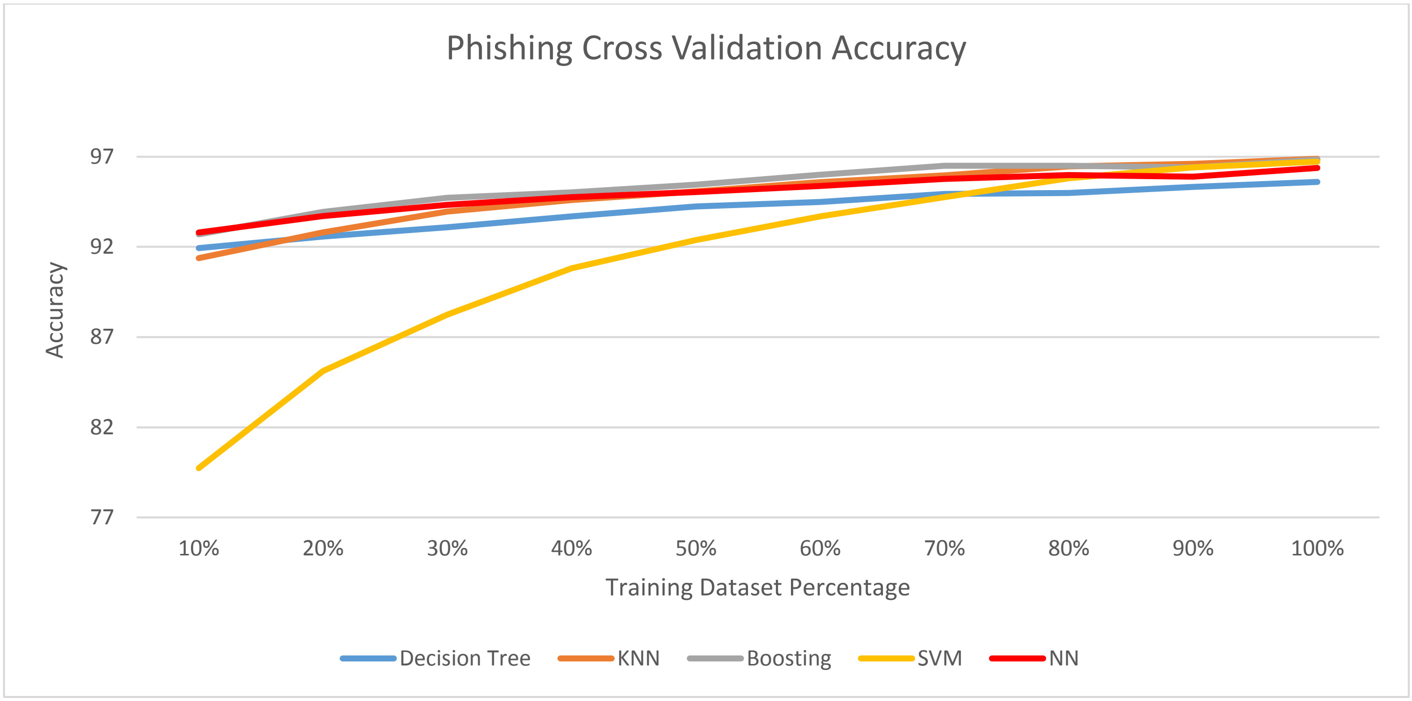
Comparison of different classifiers:

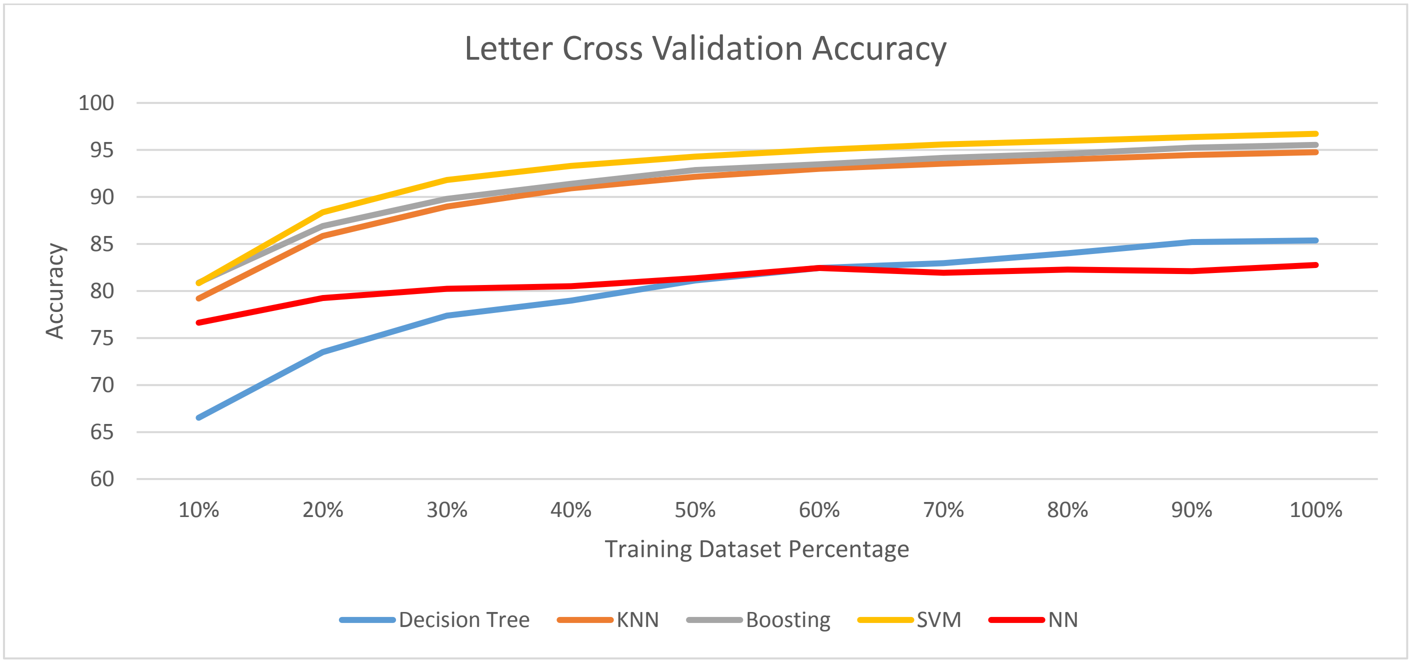
Phishing Website

	Training Accuracy	10 Fold CV Accuracy	Test Accuracy
Decision Tree	97.7511	95.605	96.2327
K-Nearest Neighbor	99.134	96.88498451	96.8957
Boosting	99.134	96.8463	96.9861
SVM	99.0953	96.72996	97.1368
Neuron Networks	99.0048	96.381	96.1724

Letter Recognition

	Training Accuracy	10 Fold CV Accuracy	Test Accuracy
Decision Tree	95.7324	85.37418	86.8574
K-Nearest Neighbor	100	94.76009	95.841
Boosting	100	95.5394	96.2236
SVM	99.2637	96.7189	97.4048
Neuron Networks	84.0946	82.765	82.1161





The phishing website dataset has achieved good performance under different classifiers and there is no significant difference among all of them. The dataset is generalized mostly by KNN classifier and it takes less time to train since KNN is a lazy learner. For Letter recognition dataset. SVM achieves the best performance but KNN and boosting has similar scores as well. Neuron Networks however, performs the worst, suggesting that the model overfits the dataset.

The cross validation curve for phishing website suggests that SVM learns very fast and quickly converges to a high accuracy value while for letter recognition dataset, decision tree and neuron networks does not perform well. The line is essentially flat for neuron networks, indicating it does not learn much with the increasing training data.

Based on Markam's table [4], the following table compares the 5 classifiers in the experiment:

Algorithm	**Results interpretable?**	**Average predictive accuracy**	**Training speed**	**Prediction speed**	**Amount of parameter tuning needed (excluding feature selection)**
KNN	Yes	Lower	Fast	Depends on K	Minimal
Decision trees	Somewhat	Lower	Fast	Fast	Some
SVM	A little	Higher	Slow	Fast	Some
AdaBoost	A little	Higher	Slow	Fast	Some
Neural networks	No	Higher	Slow	Fast	Lots

References

1. Phishing Website dataset. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Phishing+Websites

2. Letter Recognition dataset. UCI Machine Learning Repository.

https://archive.ics.uci.edu/ml/datasets/Letter+Recognition

3. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

4. Kevin Markam. Comparing supervised learning algorithms. http://www.dataschool.io/comparing-supervised-learning-algorithms/