Supervised Learning

Brittany N. Tan
Georgia Institute of Technology, brittanytan@gatech.edu

***Abstract* - This project includes analyses of different supervised learning algorithms on two different datasets. Each dataset is separated into a training set and a testing set, with the training set being 80% of the data and the testing being 20%. The data is tested using the built-in implementations of the respective learning algorithms on the data mining software Weka.**

Datasets

Letter Image Recognition Data: This dataset consists of 20,000 instances of characters, each of which is defined by 17 different attributes—one attribute for class and 16 that describe the pixels of each character. The instances are of the capital letters in the English alphabet, based on 20 different fonts and randomly distorted. The data is sorted into 26 different classes, respectively to each letter of the alphabet.

Spam E-Mail Database: This contains 4,601 instances of e-mails, of which 1,813 were considered spam. Each e-mail instance uses 58 attributes, including the class distribution. Other attributes include frequencies of certain words within the email, frequency of characters, and data on uninterrupted series of capital letters.

These datasets were chosen because of their differences in size, attributes of classification, and number of classes. The letter recognition dataset is a large set of 20,000 instances that are classified into 26 distinct nominal classes based on 17 attributes. Adversely, the spam database is a binary classification system, which is very common in the realm of machine learning; however, these two classes are based off whether a sample of humans classified e-mails as spam. Thus, while there are many more attributes to deciding whether the instances are in one of two classes, these classifications are based off of subjective human decision, as opposed to the 26 distinct, objective classes.

Decision Trees

Both datasets were divided into a training set and a testing set, 80% and 20% of the data, respectively. The first step included finding the optimal minimum number of instances per leaf on the training sets while holding the confidence factor constant.

Figure 1 illustrates the trends of each training set's accuracy each with a set confidence factor of 25% and 10 fold cross validation as the minimum number of instances per leaf increases. For each set, the optimal number of instances per leaf was 1. This is because separating an instance of data from a node containing a large group of instances will not provide as much information gain as distinguishing an instance from a node with fewer instances.

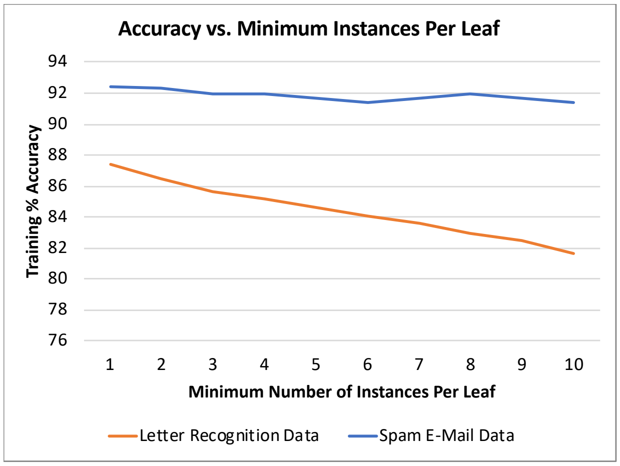

FIGURE 1
TRAINING ACCURACY BASED ON MINIMUM INSTANCES PER LEAF

Given this optimal minimal instances per leaf, the next step is to find the optimal confidence factor on the training set. This confidence threshold is representative of how much error is allowed during pruning [1]. Lowering this confidence factor means more pruning and usually more generalization in the model. Table I and Table II show the changes in training accuracies based on the confidence factor.

TABLE I
LETTER RECOGNITION TRAINING SET - CONFIDENCE VS. ACCURACY

Confidence Factor	Training % Accuracy	Number of leaves	Tree size	Time
0	87.3788	1863	3725	1.25 s
0.125	87.2788	1562	3123	1.44 s
0.25	87.3601	1619	3237	1.41 s
0.375	87.3851	1643	3285	1.49 s
0.5	87.3601	1664	3327	1.55 s
0.625	87.3913	1851	3701	53.58 s
0.75	87.3913	1851	3701	69.91 s
0.825	87.3913	1851	3701	115.83 s

TABLE II
SPAM E-MAIL TRAINING SET - CONFIDENCE VS. ACCURACY

Confidence Factor	Training % Accuracy	Number of leaves	Tree size	Time
0	92.0109	213	425	0.76 s
0.125	92.5000	95	189	0.28 s
0.25	92.3913	100	199	0.25 s
0.375	92.3098	144	287	0.25 s
0.5	92.1739	167	333	0.27. s
0.625	91.9837	199	397	4.63 s
0.75	91.9837	199	397	3.49 s
0.825	91.9837	199	397	4.29 s

For the letter dataset, the accuracy increases as the confidence factor is increased and levels out at a confidence factor of 0.625. As the confidence level increases, pruning decreases, and the tree becomes taller and deeper [2]. The accuracy also levels out for the spam dataset; however, the trend is opposite of that of the letter dataset. The reason for this is because the spam dataset has many more attributes and is also creating smaller decision trees; therefore, all the attributes in this dataset may not be directly pertinent in classifying the data.

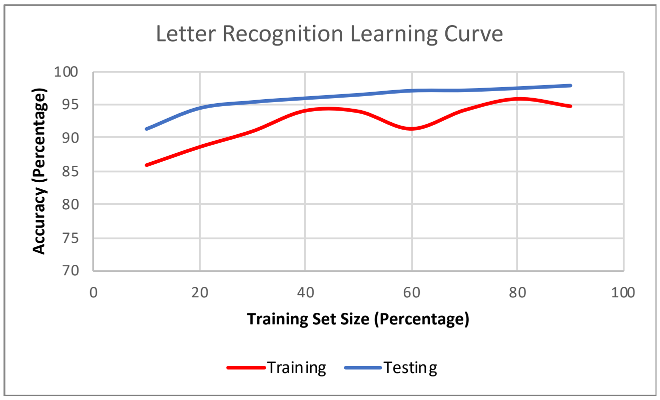

FIGURE 2
LETTER RECOGNITION DATASET - DECISION TREE LEARNING CURVE

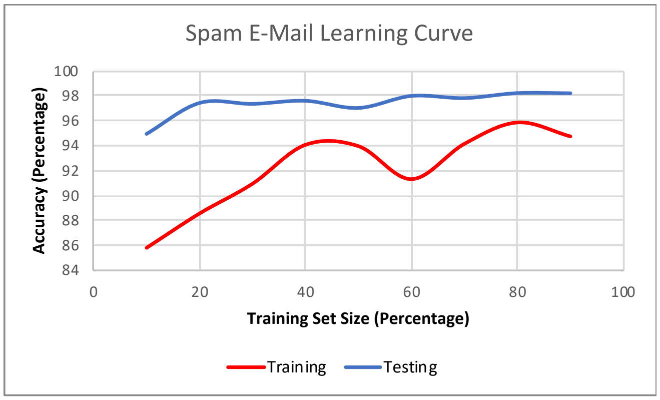

FIGURE 3
SPAM E-MAIL DATASET - DECISION TREE LEARNING CURVE

Figure 2 and Figure 3 display learning curves for the each dataset, a function of training and testing accuracy with respect to different training set sizes. Both datasets were run with their respective optimal parameters as found earlier. The gap between each curve for the letter recognition dataset is small, which is ideal for a learning curve. Additionally, the accuracy increases as the training set size increases, which indicates that the algorithm continues to generalize to new data [3]. The spam e-mail set, however, indicates high variance in its pattern due to the larger difference between training and testing accuracies. This is likely because, as mentioned before, there exist many attributes that are not pertinent in decision making. This indicates that the algorithm is overfitting, and having a larger sample of instances may help.

Another notable observation is that the time increases significantly as the confidence interval increases. This is because as pruning increases and new nodes are created per node, the same happens for each new node. This can be compared to a N-squared order function.

NEURAL NETWORKS

For this experiment, each dataset was run with varying numbers of nodes per hidden layer, then with the optimal amount from experimentation, the number of epochs to train through was altered.

TABLE III
LETTER RECOGNITION DATASET - HIDDEN LAYER NODES VS. ACCURACY

Epochs	Training %	Testing %	Cross Validation %
100	74.81	72.525	74.145
500	75.435	72.925	74.175
750	75.31	73.225	74.325
1000	75.05	73.05	74.375
1500	75.115	73	74.35

Changing the number of nodes per hidden layer in the letter recognition dataset had a drastic effect on both the training and testing accuracies. As the number of nodes per hidden layer increases, the accuracy increases. For the purposes of this experiment, the number of nodes used for testing ranged from 1 to 10, as shown in Table III. Based on the results, 10 nodes per hidden layer were used to run more tests, this time varying the number of epochs through which to train. These results are shown below in Table IV.

TABLE IV
LETTER RECOGNITION DATASET - EPOCHS TRAINED THROUGH VS. ACCURACY

Hidden Layer Nodes	Training %	Test %	Cross Validation %
1	14.56	14.075	14.665
2	33.145	34.825	32.02
3	48.405	42.55	46.58
4	57.395	55.625	62.195
5	62.805	62.375	62.195
8	72.1	70.95	71.15
10	75.435	72.925	74.175

The change in the number of epochs run through, however, did not have as drastic of an effect on the accuracy as the change in hidden layer nodes. This is likely due to the algorithm overfitting with this range of epochs.

For the spam e-mail dataset, the change in number of hidden layers did not have a significant effect on the accuracy, indicating that overfitting is happening even with the change of hidden layer nodes, contrary to the letter data. As shown in Figure 4, there does not exist any trend in neither the training nor testing accuracies when changing the number of nodes per hidden layer. When changing the number of epochs to train through, however, the testing accuracy experiences a rapid increase, and then plateaus. This is visible in the graph displayed in Figure 4. The slight dip between 600 and 800 epochs suggests that the algorithm begins to overfit there, which then persists. This is an indication that some of the instances in this dataset may be trivial; therefore, the algorithm is no longer learning from it [3].

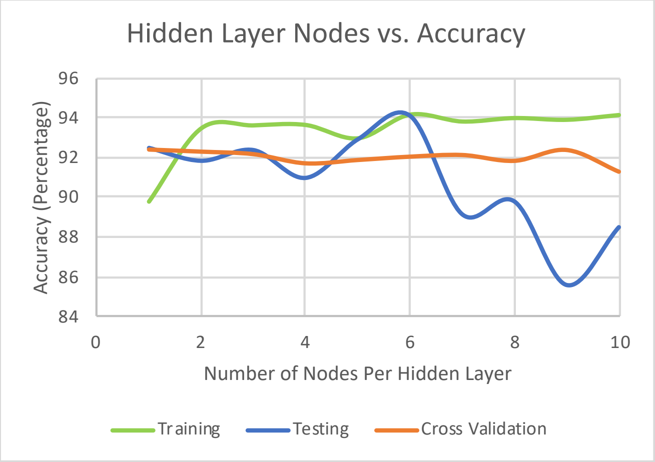

FIGURE 4
SPAM E-MAIL DATASET - HIDDEN LAYER NODES VS. ACCURACY

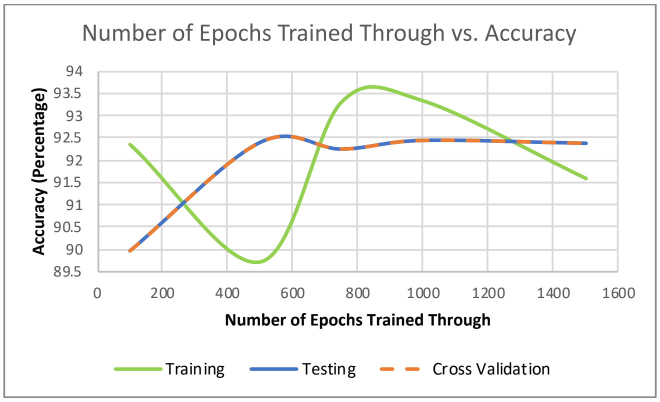

FIGURE 5
SPAM E-MAIL DATASET - EPOCHS TRAINED THROUGH VS. ACCURACY

Figure 6 and Figure 7 display the learning curves for the letter recognition dataset and the spam e-mail dataset, respectively. For the letter dataset learning curve, the testing and training accuracies remain fairly consistent as the training set size increases, and there is low difference between the two curves. The consistency in the curves indicates that the model may not have much to learn after adding more instances; therefore, many of the instances may not be directly relevant in classification.

A notable observation for the spam learning curve is that the training curve is similar to that of the letter training curve, but there is more variability in the testing curve. This shows variance in the model, and suggests that some attributes may be trivial.

This algorithm took the longest to run of the five algorithms analyzed in this project. Because neural networks include certain methods such as backpropagation and backtracking, the algorithm is constantly alternating between stepping forward and backward.

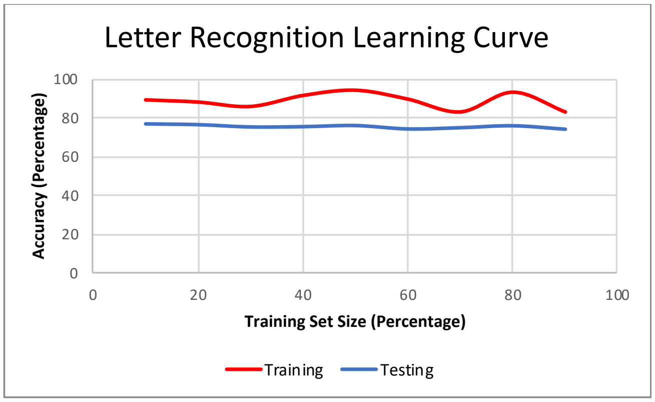

FIGURE 6
LETTER RECOGNITION DATASET - NEURAL NETS LEARNING CURVE

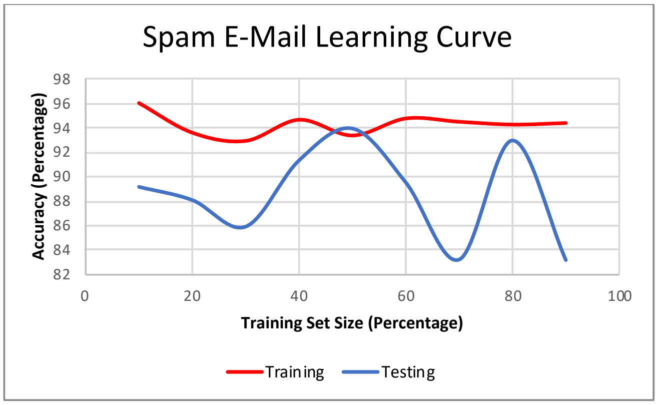

FIGURE 7
SPAM E-MAIL DATASET - NEURAL NETS LEARNING CURVE

BOOSTING

For testing this algorithm, both datasets were run using the same decision tree algorithm as mentioned before in the Decision Tree section. The hyperparameters adjusted for this experiment were the confidence factor and number of iterations. Results are shown for the letter dataset and the spam dataset in Table V and Table VI, respectively.

One significant observation to make is that neither the confidence factor or the number of boosting iterations had any effect on the training accuracy, and that the training accuracy is very high—100% for letters and near 100% for spam—no matter the magnitude of these parameters. This is indication of severe overfitting in both models.

For both datasets, increasing the number of boosting iterations increases the testing accuracy and when performing cross validation on 10 folds, the increasing trend persist. An increase in confidence factor has the same effect on the datasets as it did when performing the same decision tree algorithm without boosting. Overall, the results with boosting end up being more accurate than without boosting. This is likely because AdaBoosting generally lessens the amount of wrong classifications [4].

TABLE V
LETTER RECOGNITION DATASET - ACCURACY BASED ON BOOSTING

Confidence	Iterations	Training %	Test %	Cross Validation %
0.125	10	100	95.425	93.375
0.125	20	100	96.325	96.505
0.125	30	100	96.725	96.795
0.125	40	100	96.575	97.03
0.125	50	100	96.85	97.175
0.25	10	100	95.50	95.535
0.25	20	100	96.425	96.54
0.25	30	100	96.225	96.785
0.25	40	100	96.475	96.955
0.25	50	100	96.55	96.96
0.5	10	100	95.275	95.55
0.5	20	100	96.30	96.44
0.5	30	100	96.725	96.765
0.5	40	100	96.85	96.945
0.5	50	100	97.05	96.925

TABLE VI
SPAM E-MAIL DATASET - ACCURACY BASED ON BOOSTING

Confidence	Iterations	Training %	Testing %	Cross Validation %
0.125	10	99.9348	94.6739	95.1532
0.125	20	99.9348	95.3261	95.1967
0.125	30	99.9348	95.6522	95.1967
0.125	40	99.9348	95.5435	95.1967
0.125	50	99.9348	95.4348	95.1967
0.25	10	99.9348	95.000	95.088
0.25	20	99.9348	95.8696	95.1098
0.25	30	99.9348	95.7609	95.088
0.25	40	99.9348	95.7609	95.1315
0.25	50	99.9348	95.4348	95.1315
0.5	10	99.9348	94.3478	94.9794
0.5	20	99.9348	95.5435	95.0011
0.5	30	99.9348	95.7609	95.0011
0.5	40	99.9348	95.3261	95.0011
0.5	50	99.9348	95.9783	95.0011

SUPPORT VECTOR MACHINES

Support Vector Machine is an algorithm that forms a hyperplane that separates instances into classes [5]. In order to test this algorithm, the data was run using two different kernels, the Polynomial kernel for which the polynomial was changed, and the Radial Basis Function (RBF) kernel where the gamma value was changed. These parameters were altered to test which values produce the hyperplane with the best generalization capacity [6]. Results are shown for each dataset in Table VII and Table VIII.

For the letter recognition dataset, the accuracy increases when both the exponent increases in the Polynomial kernel and when the gamma value increases in RBF. The model experiences very drastic increases with the increase of the gamma factor in RBF, which is an inverse function of the dataset's expected variance [7], which indicates that it is possible that the model matches the function provided by this kernel. However, the strong positive correlation between the exponent and the accuracies indicates that the model may be a polynomial pattern, and because the overall accuracies are higher, this kernel is more likely to be a match for the dataset.

The spam e-mail dataset acts similarly in response to the increasing gamma factor in the RBF kernel, but decreases in accuracy when increasing the exponent in the Polynomial kernel. Based on this data, the Polynomial kernel is not a good fit for classifying this data, but the RBF kernel is.

TABLE VII
LETTER RECOGNITION DATASET - SVM ACCURACIES

Kernel	Exponent	Training %	Testing %	Cross Validation
Poly	1	83.11	81.425	82.34
Poly	2	89.85	88.05	88.345
Poly	3	95.155	93	93.295
Kernel	**Gamma**	**Training %**	**Testing %**	**Cross Validation**
RBF	0.01	57.23	48.95	55.715
RBF	0.5	87.615	85.5	86.555
RBF	1	91.735	89.375	90.395

TABLE VIII
SPAM E-MAIL DATASET - SVM ACCURACIES

Kernel	Exponent	Training %	Testing %	Cross Validation
Poly	1	90.7629	89.4565	90.4151
Poly	2	81.7648	81.1957	81.5692
Poly	3	69.724	70.2174	69.9196
Kernel	**Gamma**	**Training %**	**Testing %**	**Cross Validation**
RBF	0.01	75.9835	72.7174	74.5707
RBF	0.5	91.9583	90.7609	91.2845
RBF	1	92.9146	91.9565	92.393

K-NEAREST NEIGHBORS

Both datasets were run with training sets and testing sets at varying amounts of nearest neighbors. The unweighted results based on the change of nearest neighbors are shown below for the letter dataset and spam dataset in Table IX and Table X, respectively. Cross validation was performed with 10 folds.

TABLE IX
LETTER RECOGNITION DATASET - UNWEIGHTED

kNN	Training %	Test %	Cross Validation
1	99.9348	90.0000	90.7846
5	93.1102	90.4348	90.4151
10	91.0889	88.587	89.2197
15	90.5455	88.587	88.6981
20	89.7196	86.9565	88.1113

TABLE X
SPAM E-MAIL DATASET - UNWEIGHTED

kNN	Training %	Test %	Cross Validation
1	100	95.600	96.03
5	97.77	94.600	95.515
10	96.805	93.950	94.72
15	95.97	93.650	94.105
20	95.27	92.550	93.405

An increase in the amount of nearest neighbors means an increase in cluster size and therefore a decrease in total amount of clusters. If two or more neighbors are close in distance, it can be more difficult for the algorithm to assign an instance to a class. This results in more generalization of the data, which explains why testing accuracy decreases as the amount of nearest neighbors increases. This trend of decreasing persists in the letters dataset when applying a weight of 1/distance; however, when applying this same weight on the spam dataset, the testing accuracy increases as the amount of nearest neighbors increases. The results after applying the weight are shown below in Table XI and Table XII.

TABLE XI
LETTER RECOGNITION DATASET - WEIGHTED (1/DISTANCE)

kNN	Training %	Test %	Cross Validation
1	100	95.600	96.03
5	100	95.575	96.08
10	100	95.225	95.57
15	100	94.775	95.01
20	100	94.000	94.465

TABLE XII
SPAM E-MAIL DATASET - WEIGHTED (1/DISTANCE)

kNN	Training %	Test %	Cross Validation
1	99.9348	90	90.7846
5	99.7175	90.9783	91.5453
10	99.5218	90.9783	91.1976
15	99.3697	91.3043	91.002
20	99.1958	91.1957	91.1758

When generating the learning curve for the letter recognition dataset, the training accuracy remains at 100%, no matter the size of the training set. This indicates a severe degree of overfitting. The testing accuracy increases as the training set size increases, but the increase is not significant, as it begins at a fairly high accuracy. This is indicative of high bias in the model [3]. Figure 8 displays the learning curve for the letter recognition dataset, and Figure 9 displays the spam data learning curve.

This learning curve is similar to the aforementioned letter recognition curve in that there is evidence of severe overfitting and high bias, based on the same reasons as mentioned before. Additionally, because all the testing accuracies are so close together and do not adopt a strong trend, much of the data may be trivial.

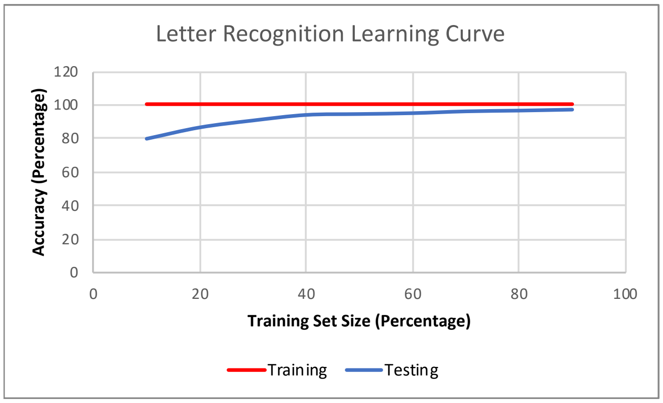

FIGURE 8
LETTER RECOGNITION DATASET - KNN LEARNING CURVE

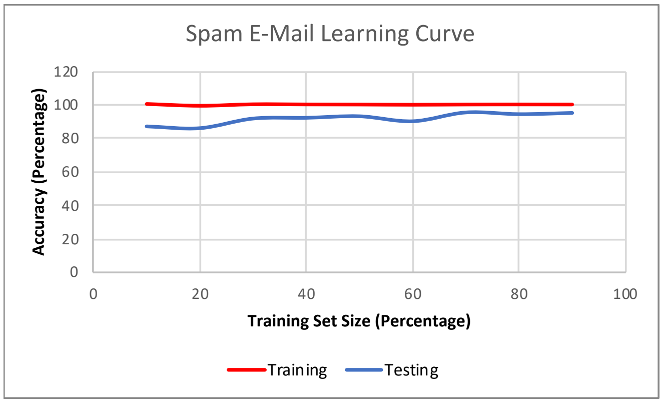

FIGURE 9
SPAM E-MAIL DATASET - KNN LEARNING CURVE

Another noticeable observation when running these tests was the very short time it took, compared to the running time of the other algorithms. The running time of these experiments also did not change in any significant magnitude with the increases in sample size, as expected from a "lazy" learning algorithm such as this one.

REFERENCES

[1] Stiglic, G., Kocbek, S., Pernek, I. and Kokol, P. (2012). Comprehensive Decision Tree Models in Bioinformatics. *PLoS ONE*, [online] 7(3), p.e33812. Available at: https://www.ncbi.nlm.nih.gov/ [Accessed 30 Jan. 2018].

[2] Bradford J.P., Kunz C., Kohavi R., Brunk C., Brodley C.E. (1998) Pruning decision trees with misclassification costs. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg [Accessed 1 Feb. 2018].

[3] R. Ng, "Learning Curve", *Machine Learning*, 2018.

[4] Rätsch, G., Onoda, T. & Müller, KR. Machine Learning (2001) 42: 287. https://doi.org/10.1023/A:1007618119488

[5] S. Patel, "Chapter 2 : SVM (Support Vector Machine)—Theory", *Machine Learning 101*, 2018.

[6] Adankon M., Cheriet M. (2009) Support Vector Machine. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA

[7] P. Jaganathan, N. Rajkumar and R. Kuppuchamy, "A Comparative Study of Improved F-Score with Support Vector Machine and RBF Network for Breast Cancer Classification", *International Journal of Machine Learning and Computing*, pp. 741-745, 2012.