Online Learning and Active Learning: A Comparative Study of Passive-Aggressive Algorithm With Support Vector Machine (SVM)

K.I Ezukwoke
University Jean Monnet, Saint-Etienne

S.J Zareian
University Jean Monnet, Saint-Etienne

Passive aggressive online learning is an extension of Support Vector Machine (SVM) to the context of online learning for binary classification. In this paper we consider the application of the algorithm on anomaly labeling for IJCNN 2001 Neural Network Competition dataset from LibSVM dataset repository[1] from Ford Research Laboratory. We also work on an improved version of the online learning algorithm called Active learning and we compare both algorithms to that of SVM (from LibSVM library). We propose different experimental setups for comparing the algorithms.

Keywords: passive aggressive online learning, active learning, Support Vector Machine (SVM)

INTRODUCTION

Online learning is a classification algorithm which is an extension of Support Vector Machine (SVM). Unlike SVM where we update the weights using batches of data samples or all data samples, here we only update the weights using one example at a time. At every iteration, we only update the weights associated with the data sample until we reach the last index.[2]

BINARY CLASSIFICATION

Binary classification is a classification problem involving only two defined categories. Given set of datasamples $\{x_i, y_i\}^N$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ where $\mathcal{Y} \in \{-1, +1\}$ for a binary classification problem.

Passive-Aggressive Online Algorithm

Passive-Aggressive online algorithm is a family of first-order online learning algorithm. We present the optimization problem of the Passive Aggressive (PA) online algorithm from Cramer et al. in [1]. Online binary classification takes place in a sequence of rounds.

On each round the algorithm observes an instance and predicts its label to be either +1 or −1. After the prediction is made, the true label is revealed and the algorithm suffers an instantaneous loss which reflects the degree to which its prediction was wrong.

At the end of each round, the algorithm uses the newly obtained instance-label pair to improve its prediction rule for the rounds to come [1].

We denote the symbols used to present the algorithm as follows. At round *t* we select an example from an independent and identical distributed random variable *xt* with label $y_y \in \{-1, +1\}$. Based on the loss suffered we compute the update rule τ and update the weight **w** at every round until we reach the final index. PA algorithms aggressively make an update whenever the loss is non-zero (even if the classification is correct) [2].

$$\begin{cases} \arg\min\limits_{\mathbf{w}\in\mathbb{R}^N} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 \\ \text{s.t} \quad y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1 \end{cases} \tag{1}$$

where w is the weight update. By introducing the Lagrangian multipliers we have

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big) \tag{2}$$

Observing the KKT optimality conditions we have

$$\nabla\mathcal{L} = 0 \tag{3}$$

So

$$\nabla_{\mathrm{w}}\mathcal{L} = \mathbf{w}-\mathbf{w}_t-\tau y_t x_t \tag{4}$$

$$\mathbf{w} = \mathbf{w}_t + \tau y_t x_t \tag{5}$$

Substituting equation (5) into (2) we have

$$\mathcal{L}(\tau) = \frac{1}{2}\tau^2 y_t^2\|\mathbf{x}_t\|^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big) \tag{6}$$

given that $y \in \{-1, 1\}$ we can rewrite the above as

$$\mathcal{L}(\tau) = \frac{1}{2}\tau^2 \parallel \mathbf{x} \parallel^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big) \tag{7}$$

$$\mathcal{L}(\tau) = -\frac{1}{2}\tau^2 \parallel \mathbf{x} \parallel^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big) \tag{8}$$

By differentiating with respect to τ we have

$$\nabla_\tau\mathcal{L}(\tau) = -\frac{1}{2}\tau^2 \parallel \mathbf{x} \parallel^2 + \tau\big(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)\big) = 0 \tag{9}$$

where

$$\tau = \frac{1-y_t(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}\|^2} \tag{10}$$

The above equation is the hard margin formulation for passive aggressive algorithm. By introducing an error term or slack variable, we can transform the hard margin formulation to a soft margin. We write the soft margin formulation of passive aggressive algorithm as

$$\begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^N,\xi\geq 0} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\sum_{i=1}^{N}\xi \\ \text{s.t } y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1-\sum_{i=1}^{N}\xi \\ \qquad \xi \geq 0 \end{cases} \tag{11}$$

Following the same procedure for deriving the updated τ as in the case of the hard margin; we arrive at the closed form solution

$$\tau = min\left\{C, \frac{1-y_t(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}\|^2}\right\} \tag{12}$$

C is the positive parameter to balance the tradeoff between **passiveness** and **aggressiveness.**

Finally we can also have a quadratic objective function with the following optimization problem

$$\begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^N,\xi\geq 0} \frac{1}{2}\| \mathbf{w}-\mathbf{w}t \|^2 + C\sum_{i=1}^{N}\xi^2 \\ \text{s.t } y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1-\sum_{i=1}^{N}\xi \\ \xi \geq 0 \end{cases} \tag{13}$$

which has the closed form solution for the τ update

$$\tau_t = \frac{l_t}{\|x_t\|^2+\frac{1}{2C}} \tag{14}$$

ALGORITHM 1
ONLINE PASSIVE-AGGRESSIVE ALGORITHM

Input : $\mathbf{X}, y, C \geq 0$
Output : $\mathbf{w}$

begin
  $\mathbf{w} \leftarrow \mathbf{w}^0$;
  for $t = 1, 2, \ldots N$ **do**
    receive instance: $\mathbf{x} \in \mathbb{R}^n$;
    predict $\hat{y} \leftarrow sign(\mathbf{w}_t \cdot \mathbf{x})$;
    correct label: $y_t \in \{-1, 1\}$;
    loss
     $l_t \leftarrow max(0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}))$;
    compute τ_t;
    update: $\mathbf{w}_t \leftarrow \mathbf{w} + \tau y_t \mathbf{x}_t$
  end
end

where τ takes the different form

1. $\tau = \frac{l_t}{\|x_t\|^2}$

2. $\tau = min\left\{C, \frac{l_t}{\|x_t\|^2}\right\}$

3. $\tau = \frac{l_t}{\|x_t\|^2 + \frac{1}{2C}}$

Active Learning

The labeling of training examples is an expensive and time consuming task in a supervised classification setting. Active learning (AL) modelling is a semi-supervised strategy that helps to reduce labelling cost by selecting the most useful unlabelled examples to train a predictive model [3]. Reducing labelling cost helps to construct a high performing classifier which keeps the amount of supervision to a minimum by selecting since it only selects the valuable training instances [4].

The promise of AL is that by iteratively increasing the size of our carefully selected labeled data, it is possible to achieve similar (or greater [5]) performance to using a fully supervised data-set with a fraction of the cost or time that it takes to label all the data. AL is considered to be a semi-supervised method, between unsupervised and fully supervised in terms of the amount of labeled data, i.e. for unsupervised data we use 0% labeled samples and for fully supervised we use 100% labeled samples. Therefore, the decision of how much data to use or alternatively how much performance is required from the model relies on a resource management decision, in other words it can be a business decision.

The active learning version of the algorithm 1's objective is to minimize the number of labels to query using a probabilistic criterion (Bernoulli random distribution). This way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data.

ALGORITHM 2
ACTIVE LEARNING VERSION OF ALGORITHM 1

Input : $\mathbf{X}, y$
Output : $\mathbf{w}$
begin
  $\mathbf{w} \leftarrow \mathbf{w}^0$;
  for $t = 1, 2, \ldots, N$ **do**
    receive instance: $\mathbf{x}_t \in \mathbb{R}^n$;
    Let $\hat{p} \leftarrow = \mathbf{w}_t \cdot \mathbf{x}$;
    predict $\hat{y} \leftarrow sign(\hat{p})$;
    Draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + |\hat{p}|)$;
    if $Z_t = 1$ **then**
      Get label $y_t \in \{-1, 1\}$;
      Loss $l_t \leftarrow max(0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}))$ compute τ_t;
      update: $\mathbf{w}_t \leftarrow \mathbf{w} + \tau y_t \mathbf{x}_t$
    else
      $\mathbf{w}_t \leftarrow \mathbf{w}$
    end
  end
end

KERNEL METHODS

Kernel Passive-Aggressive Online Learning Algorithm

Kernel methods introduces non-linearity into our model by projecting our data into a Hilbert space (H) (kernel space defined by mercer's inner product). We implement the kernel passive-aggressive (PA) algorithm and its active version (APA) by considering that any classifier can be defined as a weighted sum of seen examples. [3 4]

The kernel algorithm for passive-aggressive online algorithm is found below.

ALGORITHM 3
ONLINE KERNEL PASSIVE-AGGRESSIVE ALGORITHM

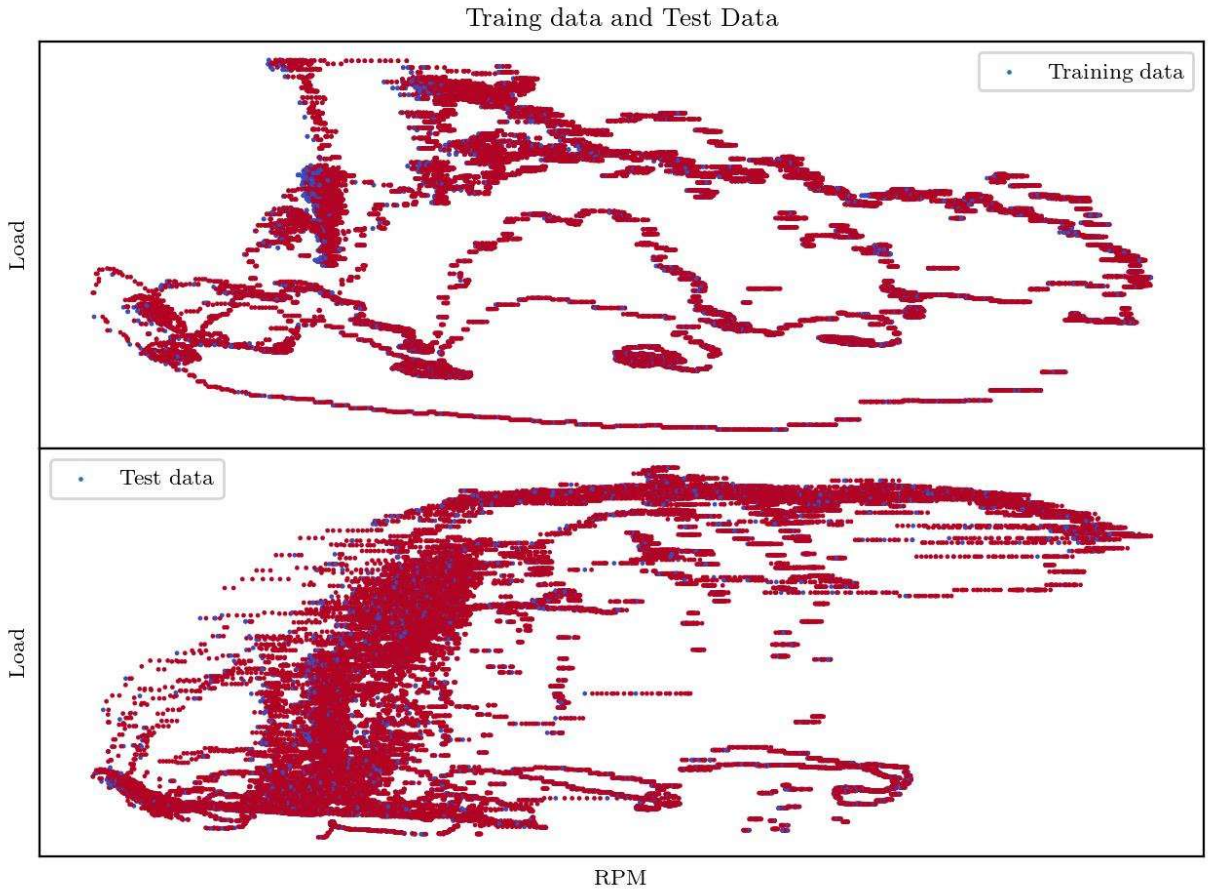
Input : $\kappa, y, C \geq 0$
Output : α
begin
  $\mathbf{w} \leftarrow \mathbf{w}^0$;
  for $t = 1, 2, \ldots N$ **do**
    receive instance: $\kappa(\mathbf{x_t}, \mathbf{x})$;
    predict $\hat{y} \leftarrow sign(\sum_{i \in \mathcal{S}} \alpha \kappa(\mathbf{x_t}, \mathbf{x}))$;
    Get correct labels: $y \in \{-1, +1\}$;
    Get support sets: $\mathcal{S} \leftarrow \mathcal{S} \cup \{t\}$;
    $\alpha_t \leftarrow y \min\{C, \frac{1 - y_t \hat{y}}{\kappa(x_t, x_i)}\}$
  end
end

EXPERIMENT

Dataset

We used ijcnn1 dataset from LIBSVM. These samples of data are produced by physical system (10-cylinder internal combustion engine). This time series dataset is labeled for sensors so that its normal firing is labeled by −1 and the misfire is labeled by +1, with normals dominating. The models we introduce here try to do anomaly detection on this time series data.

FIGURE 1



The data is categorized into three different parts as training data, validation data and test data. The training data is a combination of training and validation datasets totalling 64980 datapoints. The training data is also composed of 58689 positive triggers (with labels -1) and 6291 anomaly triggers (with labels +1). The test data has a total 91701 data points with 82989 normal triggers and 8712 abnormal triggers. In general, each sample has 27 features however, we removed unneeded features and diminished to 12 features.

Experimental Setup

We consider an experiment setup for comparing passive-aggressive online algorithm and support vector machine. We also compare active version of PA algorithm with SVM.

1. Train PA, APA and SVM using all training data (composed of 64980 datapoints) and test on 91701 data points.
2. Train model using k-fold cross-validation (precisely 10-fold cross-validation).
3. Introduce minimal noise by randomly flipping the training labels.

Performance Analysis

In this experiment we compare the performances of passive aggressive algorithm and active passive aggressive algorithm with linear support vector machine from **(LibSVM).** We compare precisely their F1-score and AUCROC score using a well organized table. We perform experimentation by tuning the hyper-parameter **C** for F1 and F2 relaxations in comparison with SVM.

Evaluation Metric

- AUCROC Score
  Area Under Curve-Receiver Operating Characteristics score is the performance of all classification thresholds (a models ability to distinguish between classes). It is given by the area under the curve of true positive rate against false positive rate. Where True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive with respect to all positive data, and is given by

$$TPR = \frac{TP}{TP+FN} \tag{15}$$

  and False Positive Rate corresponds to the proportion of negative data points that are misclassified as positives with respect to all negative data points and is given by

$$FPR = \frac{FP}{FP+tN} \tag{16}$$

- F1-Score
  F1-Score is the harmonic mean of precision and recall and it is given by

$$F1 - \text{ score } = \frac{2\times \text{ precision } \times \text{ recall}}{\text{precision + recall}} \tag{17}$$

  where

$$precision = \frac{TP}{TP+FP} \tag{18}$$

$$recall = \frac{TP}{TP+FN} \tag{19}$$

  TP: True Positives, **TN:** True Negatives, **FP:** False Positives, **FP:** False Negatives.

Classic, F1 & F2 Relaxation With C = 0.1, 1 and 10

We observe from Table 1 the result of using complete training dataset on the model using the C hyper-parameter. We observe that Passive-aggressive algorithm outperforms its active version (in most cases) and SVM (in all cases). The AUCROC and F1-score for PA model performs better in correctly labeling the anomally triggers than SVM (from LibSVM).

Table 2 shows the running time comparisons. We observe that PA algorithm has a faster running time than SVM therefore making it computationally inexpensive for large-scale anomally labelling. However, Active Passive-Aggressive (APA) online algorithm is fastest of them all, outperforming Support vector machine in time, and accuracy of labeling.

The experimental result from table 1 of learning with all examples for passive-aggressive algorithm, active passive algorithm and support vector machine. We observe that passive-aggressive algorithm **PA** performs better than SVM in all cases of C.

To compare Online Learning algorithm and its Active Learning version, in another point of view, the number of times Online Learning updates weight during learning is 3190 for classic update, 3065 with F1 relaxation, and 3162 with F2 relaxation. However the number of times Active Learning method made update was 3147 for classic, 3026 with F1 relaxation, and with F2 relaxation 3113. This is a clear indication of the difference between PA and APA, as APA requires only important datapoints for labeling.

Classic, F1 & F2 Relaxation With C = 0.1, 1 and 10 With 10-Fold Cross-Validation

We perform a 10-fold cross-validation as a second experimental setup and observe a reduction in the accuracy across all models. We observe a significant reduction in the AUCROC score for PA algorithm. APA and SVM however remain relatively stable on using 10-fold cross-validation technique.

We observe that APA algorithm gives different accuracies on numerous runs but becomes stable on performing a 10-fold cross-validation. This unstable property of active PA algorithm is annulled when we use this technique (cross-validation) while taking advantage of its computational speed (as seen in Table 4).

Classic, F1 & F2 Relaxation With C = 0.1, 1 After Flipping Labels

By introducing small amount of random noise through flipping of the labels, we observe that the models accuracies (AUCROC and F1 scores) is reduced. We observe from Table 5 that despite the introduction of random noise in the data, PA remains the best performing algorithm in comparison with SVM, outperforming it with the state-of-the-art results for different hyper-parameters.

Using the F1 relaxation, we observe that APA outperform PA in most cases while PA outperforms APA on F2 relaxation. To harness completely the efficiency of APA, it is important to consider the second experimental setup where we use cross-validation. This way we are guaranteed of the stability of APA across different hyper-parameter values.

TABLE 1
COMPARING AUCROC AND F-1 SCORES FOR PASSIVE-AGGRESSIVE (PA) ALGORITHM, ACTIVE PASSIVE-AGGRESSIVE (APA) ALGORITHM AND SUPPORT VECTOR MACHINE (SVM)

AUCROC Score (%) | F1-score (%)								
	Classic		F1_relax, C = 0.1		F1_relax, C = 1		F1_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA	**83.85**	**67.64**	85.62	**68.38**	**86.22**	**69.56**	83.85	67.65
APA	71.72	47.50	**88.34**	65.33	79.54	59.05	**86.82**	**68.09**
SVM	50.39	2.0	61.97	36.73	63.19	39.42	63.32	39.70
	Classic		F2_relax, C = 0.1		F2_relax, C = 1		F2_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA			**85.15**	**69.05**	85.26	**69.19**	82.52	**66.39**
APA			84.25	63.47	**87.77**	68.54	**85.87**	65.07
SVM			61.97	36.74	63.19	39.42	63.32	39.70

TABLE 2
COMPARING RUNNING TIME FOR PASSIVE-AGGRESSIVE (PA) ALGORITHM, ACTIVE PASSIVE-AGGRESSIVE (APA) ALGORITHM AND SUPPORT VECTOR MACHINE (SVM)

Classic Running Time (secs)	PA	APA	SVM	F1_Relax Running time (secs)	PA	APA	SVM	F2_Relax Running time (secs)	PA	APA	SVM
C = 0.1	0.89	**0.52**	19.19	C = 0.1	1.01	**0.53**	19.19	C = 0.1	0.95	**0.53**	19.19
C = 1	0.94	**0.53**	21.07	C = 1	0.95	**0.53**	21.07	C = 1	0.93	**0.53**	21.07
C = 10	0.89	**0.54**	30.78	C = 10	0.94	**0.53**	30.78	C = 10	0.92	**0.53**	30.78

TABLE 3
COMPARING AUCROC AND F-1 SCORES ON 10-FOLD CROSS-VALIDATION FOR PASSIVE-AGGRESSIVE (PA) ALGORITHM, ACTIVE PASSIVE-AGGRESSIVE (APA) ALGORITHM AND SUPPORT VECTOR MACHINE (SVM)

AUCROC Score (%) | F1-score (%) for 10-fold Cross-validation								
	Classic		F1_relax, C = 0.1		F1_relax, C = 1		F1_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA	69.87	28.67	65.66	24.89	65.67	24.89	65.67	24.89
APA	**71.36**	**47.50**	**81.08**	**66.34**	**69.23**	**27.57**	**82.43**	**66.37**
SVM	50.43	0.01	59.17	30.02	62.09	37.03	62.17	37.21
	Classic		F2_relax, C = 0.1		F2_relax, C = 1		F2_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA			**85.15**	**69.05**	85.26	**69.19**	82.52	**66.39**
APA			84.25	63.47	**87.77**	68.54	**85.87**	65.07
SVM			61.97	36.74	63.19	39.42	63.32	39.70

TABLE 4
COMPARING RUNNING TIME ON 10-FOLD CROSS-VALIDATION FOR PASSIVE-AGGRESSIVE (PA) ALGORITHM, ACTIVE PASSIVE-AGGRESSIVE (APA) ALGORITHM AND SUPPORT VECTOR MACHINE (SVM)

Classic Running Time (secs)		PA	APA	SVM	F1_Relax Running time (secs)		PA	APA	SVM	F2_Relax Running time (secs)		PA	APA	SVM
	C = 0.1	8.21	**4.45**	139		C = 0.1	7.69	**4.59**	138.3		C = 0.1	7.57	**4.57**	138.3
	C = 1	8.29	**4.48**	139.6		C = 1	7.58	**4.62**	146.4		C = 1	7.84	**4.6**	146.4
	C = 10	8.27	**4.41**	139.6		C = 10	7.68	**4.54**	233.5		C = 10	7.86	**4.68**	233.6

TABLE 5
COMPARING AUCROC AND F-1 SCORES (AFTER RANDOMLY FLIPPING LABELS) FOR PASSIVE-AGGRESSIVE (PA) ALGORITHM, ACTIVE PASSIVE-AGGRESSIVE (APA) ALGORITHM AND SUPPORT VECTOR MACHINE (SVM)

AUCROC Score (%) | F1-score (%) After Random Lable flipping								
	Classic		F1_relax, C = 0.1		F1_relax, C = 1		F1_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA	**83.25**	**65.16**	**86.72**	66.35	77.83	57.0	83.25	65.16
APA	81.70	60.9	85.04	**67.85**	**85.73**	**62.60**	**74.28**	**53.58**
SVM	50.24	0.1	60.87	34.13	62.45	37.74	62.70	38.29
	Classic		F2_relax, C = 0.1		F2_relax, C = 1		F2_relax, C = 10	
	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score	AUC_ROC	F1-score
PA			**84.29**	**63.28**	**86.84**	**66.39**	77.99	59.34
APA			77.98	57.51	66.49	41.97	**81.86**	**65.82**
SVM			60.87	34.13	62.45	37.74	62.70	38.29

CONCLUSION

In this paper we presented the experimental results of Passive-Aggressive (PA) Online algorithm, Active Passive-Aggressive (APA) Online algorithm and a comparison with Support Vector Machine (SVM). We apply these algorithms for anomaly classification/labeling from IJCNN 2001 Neural Network Challenge from Ford Research Laboratory.

We conclude that PA and APA algorithms outperformed SVM, achieving state-of-the-art results. Both PA and APA are computationally less expensive than SVM and can scale easily to labeling large datasets. We conclude that the number of mistakes made by Active Learning model is less than that of Online Learning model and performs better on 10fold cross-validation.

We also conclude that introducing some amount of noise by label flipping only reduces the AUCROC and F1-scores for PA algorithm significantly while scores for APA and SVM remains relatively stable.

ENDNOTES

1. IJCNN 2001 NN Competition dataset LIBSVM
2. Project source code github
3. source code for kernel PA github
4. source code for kernel APA github

REFERENCES

Bouneffouf, D., Laroche, R., Urvoy, T., Feraud, R., & Allesiardo, R., (2014). *Contextual Bandit for Active Learning: Active Thompson Sampling*. Researchgate publication.

Crammer K., Dekel O., Keshet J., Shalev-Shwartz S., & Singer Y. (2006). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, *7*, 551–585.

Hoi, S.C.H., Sahoo D., Lu, J., & Zhao P. (2018). *Online Learning: A Comprehensive Survey*, pp.13-15. arXiv:1802.02871v2[cs.LG]

Ilhan, O.H., & Amasyali M.F. (2014). Active Learning as a Way of Increasing Accuracy. *International Journal of Computer Theory and Engineering*, *6*(6), 460.

Wang, L., Hu, X., Yuan, B., & Lu, J-F. (2014). Active Learning via Query Synthesis and Nearest Neighbour Search. *Neurocomputing*.