DEFENDING ACTIVE LEARNING AGAINST ADVERSARIAL INPUTS IN AUTOMATED DOCUMENT CLASSIFICATION

Lei Pi[†]*, Zhuo Lu*[‡]*, Yalin Sagduyu*[*]*, Su Chen*[†]

[†] University of Memphis, TN 38152. Emails: {lpi,schen4}@memphis.edu
[‡] University of South Florida, Tampa FL 33620. Email: zhuolu@usf.edu
[*] Intelligent Automation Inc., Rockville, MD 20855, Email: ysagduyu@i-a-i.com

ABSTRACT

Business and government operations generate large volumes of documents to be categorized through machine learning techniques before dissemination and storage. One prerequisite in such classification is to properly choose training documents. Active learning emerges as a technique to achieve better accuracy with fewer training documents by choosing data to learn and querying oracles for unknown labels. In practice, such oracles are usually human analysts who are likely to make mistakes or, in some cases, even intentionally introduce erroneous labels for malicious purposes. We propose a risk-factor based strategy to defend active-learning-based document classification against human mistakes or adversarial inputs. We show that the proposed strategy can substantially alleviate the damage caused by malicious labeling. Our experimental results demonstrate the effectiveness of our defense strategy in terms of maintaining accuracy against adversaries.

Index Terms— active learning, document classification, security and attacks, malicious inputs.

1. INTRODUCTION

Daily routine operations in business and governments produce a large numbers of documents, which must be properly categorized or labeled, then disseminated to authorized personnel and stored in appropriate places. For example, documents in government operations may be labeled as public information or a classified level may be assigned according to national security requirements. Machine learning techniques, such as Naive Bayes classifier and Support Vector Machine (SVM) [1], have been extensively used as a vital assistance for automated document classification [2, 3].

To facilitate processing training data sets, active learning [4] has been used to achieve better accuracy with smaller training sets for document classification. The essential idea behind active learning is to let the learning system choose data to learn from and query an oracle for a label. In practice, such an oracle is usually a human analyst who is tasked to identify and classify given documents. For example, in government operations, security analysts are assigned to classify any documents into a security classification level for proper information control and dissemination.

On one hand, active learning can significantly reduce the size of training documents that are essential to train an underlying machine learning model [4, 5]. On the other hand, however, it also introduces risks that could lead to less accurate classification. Specifically, active learning usually involves the inputs from human analysts who can sometimes make mistakes. More severely, due to inside threats or account hacking, such inputs can even be malicious with intent to sabotage the entire active learning process. Many potential vulnerabilities in active learning can make such attacks possible [6]: 1) the attacker (i.e., a human analyst with malicious intent) can fabricate less significant data but appealing for the learner to choose; 2) the attacker can leverage existing machine learning vulnerabilities inherited by active learning; 3) the attacker can provide incorrect results when the learner queries for labels. Therefore, it should never be taken for granted that the inputs from human analysts are always correct, and it is critical to make active learning resilient to erroneous inputs due to human errors or malicious attacks.

In this paper, we aim at designing a robust active learning defense strategy. In particular, we focus on the scenario of SVM-based active learning under a malicious attacker that gives erroneous inputs during learning queries as SVM is an extensively-used method in classification and active learning [4,5,7–9]. Our defense strategy is to design a risk factor based mechanism to guide whether we should accept or reject the input from active learning. By examining the distance of a newly labeled document to the current separating hyperplane of the SVM model, the mechanism will decide if it is too risky to accept the input depending on whether the distance is larger than a given threshold. Our method is shown to substantially alleviate the damage caused by malicious attacks.

2. BACKGROUNDS AND RELATED WORK

In this section, we briefly introduce SVM and active learning.

2.1. SVM and Active Learning

SVM is a widely-used classification method [1] to find a hyperplane that separates the training data into desirable subsets with different categories/labels based on support vectors, which are a set of instances from the training data closest to the hyperplane. In SVM-based document classification, an instance is a feature vector representing the counting of words extracted from a document.

To perform accurate classification, SVM requires training based on a substantial number of instances with labels already given as the ground truth. However, labeling many instances for training a classifier could be cumbersome in practice. Hence, active learning [4,10] has been designed as an advanced process, in which only a subset of unlabeled instances is chosen to be labeled and added to the training set.

Active learning involves two parties: the learner (that is usually a machine to build an accurate classifier) and the oracle (that is usually a human analyst in practice), and consists of three components [4]: (f, q, X), where f is a classifier mapping a document into a label, X is the training set, and q is the query function, which chooses and returns the next instance from all unlabeled instances and query the oracle for the corresponding label. After each query, the learner updates X and returns a new classifier.

2.2. Adversarial Active Learning

Since active learning relies on oracles that are usually human analyst in practice, it is subject to common security vulnerabilities and exposed potential risks associated with or due to human analysts. A list of possible vulnerabilities were summarized in [6] with focus on the query strategies, leaving the risks due to human analysts less discussed.

As human analyst is an essential part in active learning, we have to consider the active learning scenario in a security sense that the inputs from human expert should not be trusted, but carefully examined to ensure security. During document classification, an analyst can maliciously label a document, which can be in fact hard to detect. When there are a fairly large number of malicious labels, the inaccuracy introduced to the resulting classifier will become significant enough to reduce or diminish the usability of an application. The work in [11] proved that it is not even necessary for an adversary to have a perfect knowledge of the classifier to launch such attacks against active learning. It is imperative to at least alleviate, if not completely eliminate, the damages due to malicious labeling in active learning for document classification.

3. MODELS AND DESIGN

In this section, we first present the models and then describe our design to protect active learning from malicious inputs.

To maintain simplicity without loss of generality, if a document set $\mathcal{D}$ can be separated into two disjoint subsets $\mathcal{D}^0$ and $\mathcal{D}^1$, we say a document $d \in \mathcal{D}$ is labeled 0 if $d \in \mathcal{D}^0$, and say d is labeled 1 if $d \in \mathcal{D}^1$.

3.1. Active Learning under Attacks

As aforementioned, the effectiveness of active learning relies on the outside inputs that may be manipulated by an adversary. Therefore, we focus on providing a defense strategy to combat such an attack to protect active learning from accepting malicious inputs. Specifically, as Fig. 1 shows, we consider an active learning process for document classification including a learner (i.e., the machine that performs active learning), a malicious human analyst that randomly gives malicious inputs, and queries from the learner to human analysts.

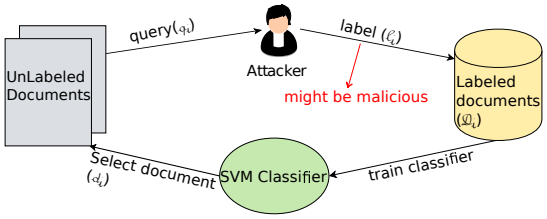

Fig. 1. The scenario of active learning under attacks.

As shown in Fig. 1, in the i-th query, the learner already has a labeled document set $\mathcal{D}_{i-1}$, and sends a query q_i containing document d_i to the analyst who then gives a (potentially wrong) label l_i to the learner. The problem is whether the learner should accept the label to form a new labeled document set $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \{d_i\}$, reject or even revert the label to keep the original labeled document set $\mathcal{D}_i = \mathcal{D}_{i-1}$.

3.2. Risk Factor based Defense Strategy

In what follows, we design a risk factor based defense strategy to protect active learning. The intuition to model the risk is as follows: in SVM, data close to a hyperplane means it is more likely to be mis-classified; if an attacker has no knowledge or access to the entire training data set, there is no way for the attacker to know where exactly the hyperplane is; therefore, the mislabeled data may have a larger distance to its hyperplane.

Consequently, if a document d_i comes from the analyst with a label l_i, we define the risk factor r_i for this document in active learning as

$$r_i = a\Delta_i^{\max}, \tag{1}$$

where $\Delta_i^{\max}$ is the maximum distance between current support vectors to the separation hyperplane based upon the existing document set $\mathcal{D}_{i-1}$, and $a > 0$ is a constant threshold.

Then, our method works as follows. When a query q_i containing document d_i is made, the learner is offered with a label l_i from the analyst. We first use the current model built upon document set $\mathcal{D}_{i-1}$ to predict the label of document d_i,

Algorithm 1 The risk-factor based defense algorithm

Given: current set $\mathcal{D}_{i-1}$, query document d_i, input label l_i
1: l_i' = $predict\_using\_current\_set(\mathcal{D}_{i-1}, d_i)$
2: if $l_i' \neq l_i$:
3: Δ_i = $compute\_distance(d_i)$
4: if $\Delta_i > r_i$:
5: return FALSE_LABEL
6: return TRUE_LABEL

Algorithm 2 Risk Factor Search Algorithm

Given: risk factor r, search step Δ, labeling error rate r_e
1: S_a = $benchmark\_test$(r, r_e, defense=False)
2: S_d = $benchmark\_test$(r, r_e, defense=True)
3: if $S_d \gg S_a$:
4: return r
5: else
6: $r = r + \Delta$
7: goto 1

and get the predicted label l_i'. If $l_i' \neq l_i$, we calculate the distance Δ_i between the d_i to current separating hyperplane, and compare it to the risk factor r_i. If $d_i > r_i$, we can think the label is mistakenly provided and reject it.

The defense process is in Algorithm 1. In algorithm 1, function $predict\_using\_current\_set$ accepts current document set $\mathcal{D}_{i-1}$ and a document d_i as parameters, and outputs the predicted label of d_i; and function $compute\_distance$ accepts the document d_i as parameter and calculates the distance between the corresponding feature vector and current separating hyperplane in the SVM algorithm.

Without doubt, this approach relies on the correctness of initial training set $\mathcal{D}_0$, which is assumed to be accurate in this paper. By design, this strategy leverages the statical property initially derived from $\mathcal{D}_0$, therefore it is not designed to prevent all attacks but only to identify and correct a subset of mislabels based on the initial and inherited statistical properties during the active learning process.

3.3. Choosing the Risk Factor

We propose to design benchmark tests on the initial training set $\mathcal{D}_0$ to adequately choose the risk factor. We used the following metric of accuracy score S to evaluate and compare the effectiveness of classification.

$$S = \frac{total\ number\ of\ accurate\ classifications}{total\ number\ of\ classifications} \qquad (2)$$

In each benchmark test (i.e, the function $benchmark\_test$ in Algorithm 2), we train the classifier using active learning with a given set of parameters, including risk factor r and defense strategy, and record the accuracy score for the testing data set as the number of queries increases. When the training set is mixed with malicious labels without any defense, the score is called affected score S_a. When the querying is protected with our defense strategy, the corresponding score is called S_d. We evaluate the effectiveness of our defense strategy by comparing S_d with S_a. Our goal is to choose a risk factor that makes the defense strategy effective, i.e., $S_d \gg S_a$. With the benchmark tests, our heuristic approach to search for the risk factor is shown in Algorithm 2.

In Algorithm 2, the labeling error rate r_e is the probability that an input is wrongly labeled in benchmark tests. In practice, it should be of small value as a large value is likely to be noticeable and raise suspicion. For example, when the government uses document analysts to classify documents, administrative approaches such as internal review and sample checking techniques can be effective in detecting such errors.

4. EXPERIMENTS AND RESULTS

In this section, we present experimental setups and results.

4.1. Experiment Setups and Parameters

We build a data set of 1264 instances with 10233 features extracted from real documents in Reuters-21578 Data Set [12]. Instances from the data set are uniformly distributed among two categories. Three fourths of the data set is used for training and the rest is for testing. Among the training set, four fifths is labeled data, the rest is the query pool.

For the SVM algorithm, we use the radial basis function (RBF) kernel with parameters $\gamma = 1.0/1264$ and $C = 1.0$. We consider three test cases: 1) *No attack*: There is no error labeling without defense strategy; 2) *Attack without defense*: There is 25% error labeling due to the attack without defense strategy; 3) *Attack with defense*: There is 25% error labeling due to the attack with defense strategy. During active learning, queries are all made randomly in the three set of tests. We first use the random sampling strategy then use the uncertainty sampling strategy in experiments [10].

4.2. Experimental Results

We first consider the scenarios that the risk factor is not carefully chosen. The query strategy in active learning is random sampling in the experiments.

Fig. 2 shows the accuracy scores of three test cases when the risk factor is too small. We can observe that the performance of the *attack with defense* case is even worse than that of the *no attack* cases under small risk factor although the number of correct inputs are 3 times more than that of malicious ones. This is because the defense strategy cannot really distinguish which input is due to the attacker, but can only detect which label may be malicious by comparing the distance of its instance in the SVM model with the risk factor. When the risk factor is too small, the defense strategy has a very

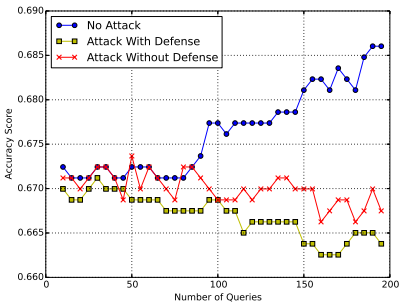

Fig. 2. Comparison of accuracy scores in three cases when $a = 0.5$.

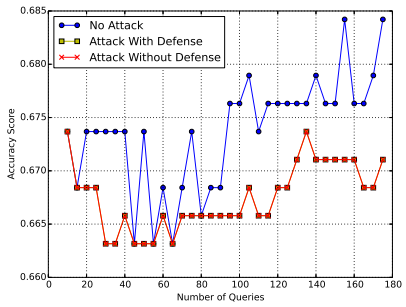

Fig. 3. Comparison of accuracy scores in three cases when $a = 1.5$.

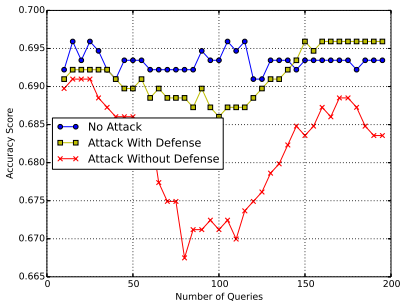

Fig. 4. Comparison of accuracy scores with optimal risk factor.

small tolerance level to accept new inputs, making the strategy erroneous by rejecting many inputs with correct labels.

Fig. 2 shows the accuracy scores of three test cases when the risk factor has a large value, i.e.; the threshold in the risk factor $a = 1.5$. As Fig. 3 depicts, the *attack with defense* case yields with the same performance as the *attack without defense* case, which is substantially worse than the *no attack* case. This means that the defense strategy neither improves nor degrades the performance of classifiers comparing with *the attack without defense* case. This is because the risk factor is chosen improperly large and the distance of each instance with error label to the separating hyperplane in the SVM classifier is considered acceptable in the defense strategy.

Figs. 2 and 3 clearly show how largely malicious inputs can affect the accuracy of document classification. The two figures also show how the value of the risk factor can affect the effectiveness of the defense strategy: a very small risk factor yields worse performance than the *attack without defense* case; and a very large risk factor leads to equal performance than the *attack without defense* case;

Then we use a risk factor that is locally optimal given in Algorithm 2. Fig. 4 shows when the risk factor is optimal, the *attack with defense* case almost achieves similar performance as the *no attack* case, and outperforms the *attack without defense* case. Admittedly, there are errors that the defense strategy cannot detect. First, it ignores mistakes where an error label is the same with the prediction of current classifier. Second, it omits the cases in which the corresponding instance of an error label is within the distance margin allowed by the risk factor. This explains why the *attack with defense* performance is overall worse than the *no attack* case. From Figs. 3 and 4, we can conclude that the defense strategy is effective when the risk factor is carefully chosen.

Finally, we evaluate the effectiveness of the defense when the query strategy is *uncertainty sampling* and compare the result with random sampling. We decrease the size of training set to half of the entire dataset and labeled set to half of training set, and increase the input error ratio to $1/2$. Fig. 5 shows in uncertainty sampling where each queried sample is

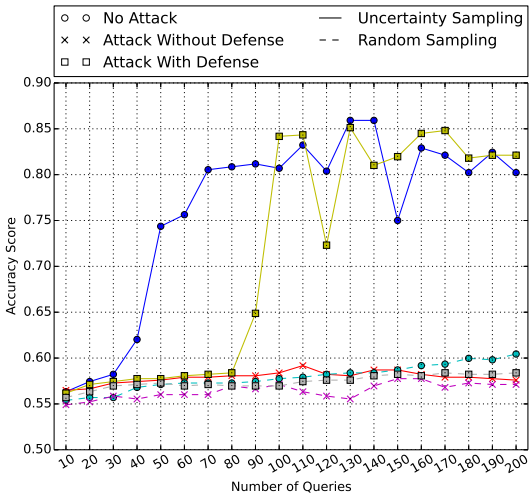

Fig. 5. Comparison of different sampling strategies.

closer to the current separating hyperplane than others, our defense is still effective in defending against erroneous labeling. With uncertainty sampling, the classifier should achieve a higher accuracy with less queries compared to random sampling. But under a heavy attack, Fig. 5 shows that the affected classifier degrades to random sampling case. With the proposed defense strategy, the damage is largely reduced and the classification accuracy is approximately equal to that of the original classifier without attack.

5. CONCLUSION

In this paper, we considered the scenario of protecting active learning in document classification against adversarial inputs. We proposed a risk-factor based defense strategy. We used real data sets and experiments to show that by adequately adjusting the risk factor, the proposed defense strategy can improve the classification accuracy and therefore shows its effectiveness in defending active-learning-based document classification against adversarial inputs.

6. REFERENCES

[1] J. Huang, J. Lu, and C. X. Ling, "Comparing naive bayes, decision trees, and svm with auc and accuracy," in *Proc. of IEEE International Conference on Data Mining (ICDM)*, 2003, pp. 553–556.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[3] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[4] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[5] E. Pasolli, F. Melgani, D. Tuia, F. Pacifici, and W. J. Emery, "Svm active learning approach for image classification using spatial information," *IEEE Trans. Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2217–2233, 2014.

[6] B. Miller, A. Kantchelian, S. Afroz, R. Bachwani, E. Dauber, L. Huang, M. C. Tschantz, A. D. Joseph, and J. D. Tygar, "Adversarial active learning," in *Proc. of Workshop on Artificial Intelligent and Security Workshop*, 2014, pp. 3–14.

[7] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for svm protein classification." in *Pacific Symposium on Biocomputing*, vol. 7, 2002, pp. 566–575.

[8] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

[9] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. of ACM International Conference on Multimedia*, 2001, pp. 107–118.

[10] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[11] D. Lowd and C. Meek, "Adversarial learning," in *Proc. of ACM International Conference on Knowledge Discovery in Data Mining (SIGKDD)*, 2005, pp. 641–647.

[12] W.-N. Hsu and H.-T. Lin, "Active learning by learning," 2015.