Identifying Gambling Websites with Co-training

Chenyang Wang ★ Pengfei Xue ★ Min Zhang ★ Miao Hu ★

★ National University of Defense Technology

Abstract

Gambling websites do great harm to society and many even cause serious network crime. To identify the gambling websites, many machine learning based methods are proposed by analysing the URL, the text, and the images of the websites. Nevertheless, most of them ignore one important information, i.e., the text within the website images. The text on the images of gambling websites has keywords that clearly point to such websites. Motivated by this, in this paper, we propose an co-training based gambling website identification method by combining the visual and semantic features of the website screenshots. First, we extract text information from webpage screenshots through the optical character recognition (OCR) technique. Then we train an image classifier based on a convolutional neural network (CNN) and a text classifier based on TextRNN respectively from image view and text view. Second, the two classifiers are retrained on unlabeled data with the co-training algorithm. Third, we conduct experiments on the webpage screenshot dataset we collected. The experimental results indicate that OCR text has strong semantic feature and the proposed method can effectively improve the performance in identifying gambling websites.

Index terms— *Co-training, Convolutional Neural Network, TextRNN.*

1 Introduction

Nowadays, most people get information from the Internet. However, the Internet is full of malicious content, especially gambling websites, which are on the edge of network crime and do great harm to the society. Due to the huge number and continuous updating of gambling websites, it is difficult to identify manually. Therefore, it is necessary to design an automatic, efficient, and accurate method to identify gambling websites.

The existing malicious website identification methods could be classified into black-list based, URL based [1, 2, 3], webpage content based [4, 5, 6, 7] and mixed-features based [8, 9]. Black-list based methods establish a black list by collecting the malicious URLs or domain names. It is labor-intensive to establish and maintain the black list, and the detection efficiency is slow. URL based methods extract features from URLs for classification. However, because the URLs contain insufficient information, the identification accuracy of URL based methods is not high. Webpage content based methods extract content features from webpages for identification, such as HTML text, image, link, JavaScript code, etc. Mixed-feature based methods combine different features to improve the accuracy of classification. For both webpage content based methods and mixed-feature based methods, when extracting visual features from images for gambling websites identification, the accuracy of them is not too high due to the high complexity of the image.

Figure 1: The text information on the website images

However, in our study, we find that there are some text information on website images that has strong semantic features and can be used to identify gambling websites. Figure 1 shows an example of a gambling website that has the text ”投注0%风险,稳赚不赔” (the text in English means that ”there is no risk of betting and you can earn money without loss”). The word ”投注” (”betting”) clearly points to gambling. In order to avoid the keyword matching detection methods, some gambling websites do not have such obvious keywords in the html text, but in the website images. There are two challenges to solve the problem: the one challenge is how to extract the text information within the images of websites. The other challenge is how to combine image and the text information to identify gambling websites.

Motivated by this, in this paper, we propose an co-training based gambling websites identification method, which extracts visual and semantic features of webpage screenshots, and utilizes unlabeled data to improve the performance of classification models. The idea of co-training is to benefit from two (or more) models which are trained from different views. These views may be obtained from multiple sources or different feature subsets (e.g., image is a view and text is another view). The two models are complementary to one another and can help correct each

DOI reference number: 10.18293/SEKE2022-106

other when they make mistakes. Naturally, the idea of co-training suits the task of learning a classification system from the image and the text view to identify gambling websites. Specifically, the proposed method is implemented as follows. Firstly, we extract text information from webpage screenshots by OCR [10] technique. Then, we train an image classifier based on CNN [11]. and a text classifier based on TextRNN [12] respectively from image view and text view. Finally, we retrain the image and text classifiers on unlabeled data with the co-training algorithm. The proposed method has the following advantages: (i) It extracts text information on website images which has strong semantic features and is useful in identifying gambling websites. (ii) It proposes a gambling websites identification method based on the multi-view semi-supervised learning algorithm (co-training) which uses unlabeled data to improve the performance of the classification model. (iii) It combines mixed-feature based identification method with semi-supervised learning to better identify gambling websites. The main contributions of this paper are as follows.

1. We propose an co-training based gambling websites identification method in this study. Compared to existing methods, this method make full use of the visual feature and key semantic feature of webpage screenshots. Through the multi-view semi-supervised learning (co-training) algorithm, this method utilizes a large number of unlabeled data to improve the performance of classification models.
2. We use OCR technique to extract the text information on webpage screenshots which has strong semantic features. Then, we train an image classifier based on CNN and a text classifier based on TextRNN respectively from image view and text view. Finally, we retrain the image and text classifiers on unlabeled data with the co-training algorithm to improve the performance of the two classifiers.
3. We evaluate the proposed method by conducting experiments in the gambling dataset we collected. The experimental results show that the proposed method can effectively improve the performance of classifiers in terms of precision, recall, and F1-score.

2 Background

Co-training. The traditional supervised learning method uses a large number of labeled data to establish a model and predict the labels of unknown instances. If only a small number of labeled data is used, the trained model is difficult to have strong generalization ability. Semi-supervised learning attempts to make the machine automatically use a large number of unlabeled data to assist a small number of labeled data in learning. The goal is to obtain the best generalization performance on these unlabeled data [13].

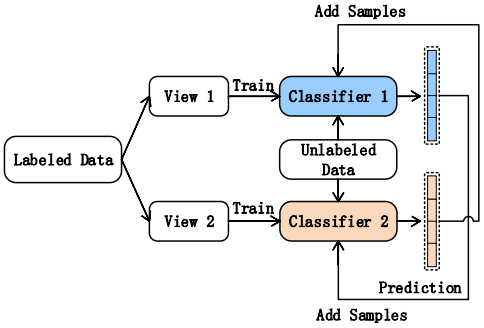


Figure 2: Base framework of co-training

Co-training [14, 15, 16] is a multi-view semi-supervised classification algorithm which improves the generalization performance of models by combining two (or more) classifiers trained from different views. Firstly, the labeled data is used to train the classifiers under different views. Then the classifiers predict on unlabeled data and label the samples with high prediction confidence. The samples that are labeled with pseudo-labels are added to the training datasets of other classifiers for retraining. Co-training is shown in Figure 2.

Different views exchange the prediction labels of unlabeled samples to realize the information exchange. The co-training algorithm is based on two key assumptions. The one assumption is that each view contains enough information to build the optimal learner. The other assumption is that the two (or more) views are independent under the condition of a given class label. Although the process of co-training algorithm is simple, the theory proves that if the two views satisfy the two key assumptions, the generalization of weak classifiers can be improved to any high level through co-training using unlabeled data. The two key assumptions are often difficult to satisfy in real tasks, so the performance improvement will not be so large. However, research shows that co-training can effectively improve the performance of weak classifiers [13]. Qiao et al. [17] present Deep Co-Training (DCT) based on co-training framework for semi-supervised image recognition, which improves the accuracy of models. Katz et al. [18] propose an ensemble-based co-training approach that make use of unlabeled text data to improve text classification when labeled data is very small.

3 Methodology

3.1 Overview of the Proposed Method

Figure 3 shows our method based on co-training for gambling website identification. The webpage screenshot data can be divided into two views: one is the image view data, and the other is the text view data on the image which can be extracted by OCR. Although the image and text data

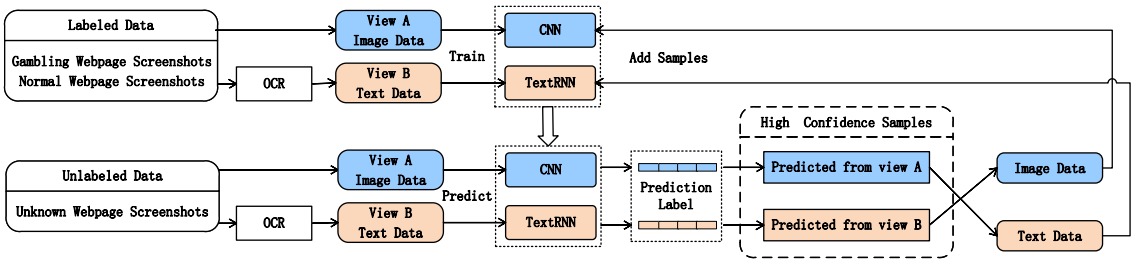


Figure 3: Overall framework of the proposed method.

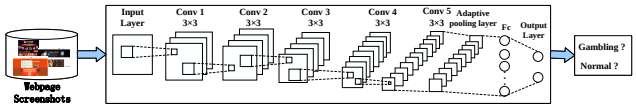


Figure 4: The image classifier based on CNN

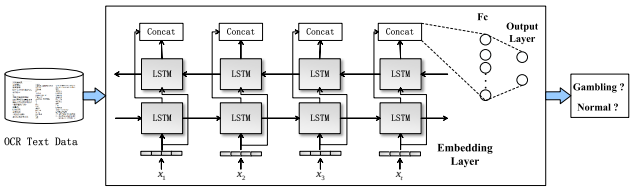


Figure 5: The text classifier based on TextRNN

are extracted from the same webpage screenshot, they are from different views, satisfying the sufficient redundancy and conditional independence of co-training.

First, we extract text messages on webpage screenshots by OCR technique. The webpage screenshots are the data of view A (image view) and the OCR texts are the data of view B (text view). We train an image classifier based on CNN and a text classifier based on TextRNN respectively from view A and view B on labeled data. Second, the unlabeled data are predicted from view A and view B. The samples with high prediction confidence are selected and labeled. The image data of the selected samples predicted from text view are added to the training dataset of the image classifier. Similarly, the text data of the selected samples predicted from image view are added to the training dataset of the text classifier. Third, the two classifiers are retrained with new training datasets.

3.2 Image Classifier based on CNN

The image classifier is constructed based on the CNN model. The input are webpage screenshots and the output of the fully-connected network are prediction labels, which are gambling or normal. The image classifier based on CNN is shown in Figure 4.

We stack 3×3 small convolutional kernels to construct convolutional networks instead of using large convolutional kernels like 7×7. Because the receptive field of a small convolutional kernel is smaller, it can extract finer-grained image features. In addition, a larger convolutional kernel requires more computation than a small one. Thus, we stack several 3×3 kernels to construct the CNN model. We add an adaptive pooling layer after convolutional layers so that the model can process the input of different sizes.

3.3 Text Classifier based on TextRNN

A text classifier is constructed based on TextRNN, the input is text data of webpage screenshots extracted by OCR technique. We use pre-trained word vectors as embedding representation, the output of hidden layer at moment t is semantic feature and input to fully-connected layer for classification. The text classifier based on TextRNN is shown in Figure 5.

We use the Sogou News Chinese corpus to train the pre-trained word vector as the embedding layer. Words in the word list are represented as word vectors, words not in the word list are represented as random vectors, and the dimension of word vectors is 300. In hidden layers we use Bi-LSTM, and each LSTM unit has the following vectors in the time step t: An input gate i_t, an output gate o_t, a forgetting gate f_t, a memory unit c_t, and hidden layer state h_t. The input gate i_t process the input of the current sequence position. The forgetting gate f_t determines what information is discarded from the cell.

3.4 Co-training of Image and Text Classifiers

First, we train image and text classifiers on labeled dataset $L = \{(x_i^A, x_i^B, y_i)|i = 1, \ldots, N\}$. Where x_i^A is the image data of sample x_i, x_i^B is the text data of sample x_i, y_i is the label, and N is the total number of samples. Second, the image classifier predicts on unlabeled dataset $U = \{(x_i^A, x_i^B)|i = 1, \ldots, M\}$, select p positive samples and q negative samples with high prediction confidence. Text data $\{(x_t^B, y_t)|t = 1, \ldots, p+q\}$ of selected samples is added to the training dataset L_B of the text classifiers. The labels $\{y_t|t = 1, \ldots, p+q\}$ are pseudo-labels. Similarly, $p + q$ samples with high prediction confidence

Algorithm 1 Co-training of image and text classifiers

Input:

Labeled dataset $L = \{(x_i^A, x_i^B, y_i)|i = 1, \ldots, N\}$,
Unlabeled dataset $U = \{(x_i^A, x_i^B)|i = 1, \ldots, M\}$,
Iteration number g,
The confidence threshold θ.

Output:

The trained image and text classifiers.

Process:

1: Train an image classifier based on CNN on labeled dataset $L_A = \{(x_i^A, y_i)|i = 1, \ldots, N\}$.
2: Train a text classifier based on TextRNN on labeled dataset $L_B = \{(x_i^B, y_i)|i = 1, \ldots, N\}$.
3: The image classifier predicts on unlabeled dataset U and selects p positive samples and q negative samples with confidence higher than θ. And these $p + q$ selected samples are added to the training dataset of the text classifier. The new dataset $L_B = \{(x_i^B, y_i)|i = 1, \ldots, N + p + q\}$.
4: The text classifier predicts on unlabeled dataset U and selects p positive samples and q negative samples with confidence higher than θ. And these $p + q$ selected samples are added to the training dataset of the image classifier. The new dataset $L_A = \{(x_i^A, y_i)|i = 1, \ldots, N + p + q\}$.
5: Remove the selected $2p + 2q$ samples from U.
6: Retrain the image classifier on the new dataset L_A.
7: Retrain the text classifier on the new dataset L_B.
8: Repeat steps 3 to 7 g times.

that predicted by the text classifier are added to the training dataset L_A of the image classifier. Third, image and text classifiers are retrained on new training datasets L_A and L_B. The above process is repeated several times, The algorithm is as Algorithm 1.

4 Experiments and Analysis

Datasets. We use crawlers to get webpage screenshots of the website on the Internet. We collect 1600 webpage screenshots, including 800 of gambling and 800 of normal (including movie, science, education, traffic, shopping, medical, etc.). These images are labeled to form the dataset L_A. We extract text data from webpage screenshots by OCR. These text data are preprocessed to form the dataset L_B. In addition, a lot of webpage screenshots are collected and used for unlabeled dataset U. A total of 600 webpage screenshots of gambling and normal websites constitute the test dataset T_A and 600 OCR text data constitute the test dataset T_B.

Evaluation metrics. In this paper, we use three evaluation metrics to evaluate the method, including Precision, Recall, and F1-score.

Table 1: Results of CNN-5 and Resnet34

Evaluation metrics	Model	Gambling	Normal	Overall
Precision	Resnet34	0.7639	0.8958	0.8299
	CNN-5	**0.8364**	**0.9111**	**0.8737**
Recall	Resnet34	0.9167	0.7167	0.8167
	CNN-5	**0.9200**	**0.8200**	**0.8700**
F1	Resnet34	0.8333	0.7963	0.8148
	CNN-5	**0.8762**	**0.8632**	**0.8697**

Table 2: Results of TextRNN on OCR text and html text

Evaluation metrics	Data	Gambling	Normal	Overall
Precision	html text	0.9052	0.9218	0.9135
	OCR text	**0.9794**	**0.9515**	**0.9654**
Rrcall	html text	0.9233	0.9033	0.9133
	OCR text	**0.9500**	**0.9800**	**0.9650**
F1	html text	0.9142	0.9125	0.9133
	OCR text	**0.9645**	**0.9655**	**0.9650**

4.1 Image Classification based on CNN

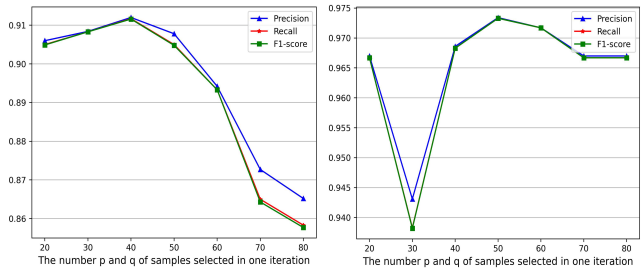
We construct an image classifier named CNN-5 and stack five 3×3 small kernels in convolutional layers as the feature extraction module. After the last convolutional layer, an adaptive pooling layer is added to unify the image output size, and then a fully-connected layer is connected as the classification module. We also do the same experiment on the pre-trained model Resnet34 for comparison. When using the Resnet34 pre-trained model, we freeze all feature extraction modules, that is, all convolution layers, and update the weight of the fully-connected network of the classification module. The two models are trained on labeled image dataset L_A. The test results of CNN-5 and Resnet34 on the test dataset T_A are shown in Table 1.

From Table 1, we can observe that the test result on Precision, Recall, and F1-score of CNN-5 is better than that of the Resnet34 pre-trained model. At the same time, because the CNN-5 model only uses five small convolutional kernels and one fully-connected layer, the training speed of the CNN-5 model is faster. We choose the CNN-5 model as the image classifier of co-training.

4.2 Text Classification based on TextRNN

When constructing a text classifier based on TextRNN, we use two bidirectional LSTM layers and one fully-connected layer. The dimension of word vectors is 300 and the hidden size of two bidirectional LSTM layers is 128. We do the text classification experiment both on OCR text extracted from the webpage screenshots and html text. The test results on the test dataset T_B are shown in Table 2.

From Table 2, we can see that the test result on Precision, Recall, and F1-score of using OCR text is better than that of using html text. The screenshots of webpage may contain some key text information that is not contained in



(a) CNN-5 (b) TextRNN

Figure 6: Performance of CNN-5 and TextRNN with different numbers of samples selected in one iteration

html. So it is necessary to extract text information from webpage screenshots by OCR. The text in the image has strong semantic features and is useful for identifying gambling websites.

4.3 The Number of Samples Selected in One Iteration

In this paper, the number of samples selected in one iteration is an important factor. In one iteration, p positive samples and q negative samples with confidence higher than threshold θ are selected. If we select a few samples, the efficiency of the algorithm is low. If we select a lot of samples, we may introduce some noisy samples with wrong labels. To obtain appropriate number p and q of samples selected, we perform the experiments with different numbers of samples. The experimental results are shown in Figure 6.

From Figure 6, we can observe that when the number p and q of samples selected in one iteration is 20, the performance of CNN-5 and TextRNN is poor, and the performance is better as the number increases. When the number is 40, CNN-5 obtain the best performance, when the number is 50, TextRNN obtain the best performance. After that, as the number increases, the performance of CNN-5 and TextRNN deteriorates. This phenomenon may be explained by the fact that when we select a small number of samples, we may lose a lot of useful samples. When we select a large number of samples, we may introduce some noisy samples with wrong labels, and the performance will decrease. When the number is within an appropriate range, we can avoid introducing noisy samples as much as possible and retain more useful samples for classification.

4.4 Co-training of CNN-5 and TextRNN

After the initial training of CNN-5 and TextRNN, the next step is to retrain the two classifiers with co-training. We set the prediction confidence threshold θ to 0.8, the number p and q of samples selected in one iteration to 50, and the iteration number g to 6. After six iterations of retraining on the unlabeled dataset U, the co-training of CNN-5 and TextRNN is completed, named Co-CNN-5 and Co-TextRNN. The test results of Co-CNN-5 and Co-TextRNN on the test datasets T_A and T_B are shown in Table 3.

Table 3: Results of CNN-5 and TextRNN after co-training

Evaluation metrics	Model	Gambling	Normal	Overall
Precision	CNN-5	0.8364	**0.9111**	0.8737
	Co-CNN-5	**0.8947**	0.9054	**0.9001**
	TextRNN	**0.9794**	0.9515	0.9654
	Co-TextRNN	0.9739	**0.9932**	**0.9835**
Recall	CNN-5	**0.9200**	0.8200	0.8700
	Co-CNN-5	0.9067	**0.8933**	**0.9000**
	TextRNN	0.9500	**0.9800**	0.9650
	Co-TextRNN	**0.9933**	0.9733	**0.9833**
F1	CNN-5	0.8762	0.8632	0.8697
	Co-CNN-5	**0.9007**	**0.8993**	**0.9000**
	TextRNN	0.9645	0.9655	0.9650
	Co-TextRNN	**0.9835**	**0.9832**	**0.9833**

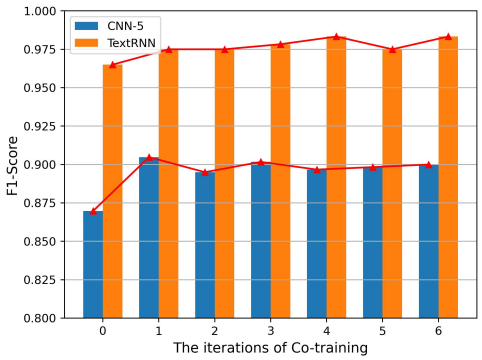


Figure 7: The performance of CNN-5 and TextRNN in different iterations of co-training

From Table 3, we can observe that after co-training, the test results on Precision, Recall, and F1-score of Co-CNN-5 and Co-TextRNN are better than that of CNN-5 and TextRNN before co-training. It indicates that the performance of the two classifiers are improved after co-training, and proves the effectiveness of the proposed method.

Meanwhile, the performance of CNN-5 and TextRNN in different iterations of co-training is shown in Figure 7. We can observe from Figure 7 that the F1-score of the CNN-5 and TextRNN shows an overall upward trend with the increase of the number of iterations. It infers that if the number of selected samples in one iteration is within an appropriate range, we can avoid introducing noisy samples and select more useful samples to improve the performance of classification models with co-training. We can observe that the performance of TextRNN is much better than CNN-5, this may be because the visual feature of webpage screenshots is too complex and the OCR text on the image has strong semantic features.

5 Related Work

URLs based methods extract features from URLs for classification. Fan et al. [1] identify illegal websites by detecting whether the unknown website contains illegal URL features. Garera et al. [2] study four different types of URL confusion structures used in phishing attacks, propose 18 URL features, and classify websites by logistic regression. Ma et al. [3] propose a website classification method based on URL features, which integrates the vocabulary features and domain name features of URLs. However, because the URLs contain insufficient features and information, the identification accuracy of URLs based methods is not high.

Webpage content-based methods extract content features from webpages for identification, such as HTML text, image, link, JavaScript code, and so on. Zhang et al. [4] extracted Chinese text from webpages and used text classification technology to classify webpages according to different themes. Li et al. [5] extract visual features from webpage screenshots to identify gambling and porn websites. Cernica et al. [6] propose a method that combines multiple techniques together with Computer Vision technique to detect phishing webpages. Jain et al. [7] propose a phishing website detection method by analyzing the hyperlinks in the webpage.

Mixed-feature-based methods combine different features to improve the accuracy of classification. Zhang et al. [8] propose a two-stage extreme learning machine for phishing website detection based on the mixed features of URL, web, and text content. Chen et al. [9] extract features from images and text in the webpages to detect gambling and porn websites.

6 Conclusion

In this paper, we propose a gambling websites identification method based on co-traning that extracts visual and semantic features of webpage screenshots and utilizes unlabeled data to improve the performance of classification models. We use OCR technique to extract text information in the webpage screenshots, the OCR text has strong semantic feature and is useful for identifying gambling websites. Then we construct an image classifier based on CNN and a text classifier based on TextRNN, the two classifiers are respectively trained from image view and text view. We retrain the image and text classifiers on unlabeled data with co-training algorithm. The experimental results indicate that the proposed method can effectively improve the performance of classifiers.

Acknowledgment

This work has been supported by the National Key R&D Program of China under Grant 2021YFB3100500.

References

[1] Y. Fan, T. Yang, Y. Wang, and G. Jiang, "Illegal website identification method based on url feature detection," *Computer Engineering*, 2018.

[2] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proceedings of the 2007 ACM workshop on Recurring malcode*, 2007, pp. 1–8.

[3] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 1245–1254.

[4] D. Zhang, "Research and implementation of content-oriented web page classification," *Nanjing University of Posts and Telecommunications, Nanjing, China*, 2017.

[5] L. Li, G. Gou, G. Xiong, Z. Cao, and Z. Li, "Identifying gambling and porn websites with image recognition," in *Pacific Rim Conference on Multimedia*. Springer, 2017, pp. 488–497.

[6] I. Cernica and N. Popescu, "Computer vision based framework for detecting phishing webpages," in *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2020, pp. 1–4.

[7] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 2015–2028, 2019.

[8] W. Zhang, Q. Jiang, L. Chen, and C. Li, "Two-stage elm for phishing web pages detection using hybrid features," *World Wide Web*, vol. 20, no. 4, pp. 797–813, 2017.

[9] Y. Chen, R. Zheng, A. Zhou, S. Liao, and L. Liu, "Automatic detection of pornographic and gambling websites based on visual and textual content using a decision mechanism," *Sensors*, vol. 20, no. 14, p. 3989, 2020.

[10] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of ocr research and development," *IEEE Computer Society Press*, 1995.

[11] T. Technicolor, S. Related, T. Technicolor, and S. Related, "Imagenet classification with deep convolutional neural networks [50]."

[12] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[13] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowledge and Information Systems*, vol. 24, no. 3, pp. 415–439, 2010.

[14] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[15] Z.-H. Zhou, M. Li *et al.*, "Semi-supervised regression with co-training." in *IJCAI*, vol. 5, 2005, pp. 908–913.

[16] Y. Wang and T. Li, "Improving semi-supervised co-forest algorithm in evolving data streams," *Applied Intelligence*, vol. 48, no. 10, pp. 3248–3262, 2018.

[17] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 135–152.

[18] G. Katz, C. Caragea, and A. Shabtai, "Vertical ensemble co-training for text classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 2, pp. 1–23, 2017.