2019 Master thesis

Threats in Malicious Domain Names and Cloud Service Abuse

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: January 29th, 2020

Advisor: Prof. Masato Uchida

Research guidance: Research on Performance Evaluation of Information Systems

Waseda University

Graduate School of Fundamental Science and Engineering

Department of Computer Science and Communications Engineering
Student ID: 5118F089-3

Naoki Fukushi

Contents

Figures

Tables

1

Introduction

1.1 Background

The Internet has become an indispensable infrastructure in the modern information society, and the convenience of our lives has greatly improved. On the other hand, the threat of cyber attacks targeting important services and confidential information on the Internet is increasing. Many cyber-attacks are caused by the attacker infecting the target device with malicious software (malware). The attacker illegally manipulates the device infected with malware, obtains or falsifies data, and performs new cyber-attacks using that device as a stepping-stone.

When the attacker performs cyber-attacks, communication on the Internet always occurs. For example, a device infected with malware called "Bot" communicates with a command and control (C&C) server and performs cyber-attacks upon receiving a command from an attacker. Similarly, both spams to direct users to malicious sites and brute-force attacks to decrypt user account/password information communicate with target devices and servers. In other words, it is important to analyze communication data on the Internet when considering effective countermeasures against cyber-attacks.

In this thesis, we focus on domain names and IP addresses (that is, address information), which are the most basic and essential identifiers of communication data on the Internet. Here, the domain name and IP address abused by the attacker are called the malicious domain name and malicious IP address, respectively. By collecting malicious domain names and malicious IP addresses, we can detect and block malicious communications.

On the other hand, attackers are making cyber-attacks more sophisticated every day, and malicious domain names and malicious IP addresses are changing accordingly. In other words, the above countermeasure is effective temporarily but not permanently. The purpose of this thesis is to take more effective countermeasures against cyber-attacks by analyzing the details of malicious domain names and malicious IP addresses and extracting their characteristics. Specifically, we propose a machine learning model that can detect malicious domain names with high accuracy by learning only a small amount of training data. Moreover, we conducted a large-scale analysis of the actual status of cyber-attacks that abuse cloud services.

1.2 Outline

The rest of this thesis is organized as follows. In Chapter 2, we propose a machine learning model that can detect malicious domain names with high accuracy by learning only a small amount of training data. In the conventional method, to detect malicious domain names with high accuracy, a large number of labeled benign/malicious domain names data is required. However, the enormous cost of correctly labeling a large number of domain names is a problem. Therefore, we proposed a labeling method based on active learning. Active learning is a machine-learning approach to select and label domain names that are expected to improve classification accuracy. As a result of the evaluation experiment using real domain names data, we realized the highly accurate detection of malicious domain names by labeling and learning a much smaller number of domain names than the conventional method. In Chapter 3, we conduct a large-scale analysis of the actual status of cyber-attacks that abuse cloud services. In the cloud service, IP addresses of the cloud service provider are shared between users. In this situation, there is a risk that legitimate users will be subjected to various restrictions if IP addresses blacklisted for malicious activity in the past are assigned to the servers they use. As a result, there is also a risk that the reputation of the cloud service provider will decrease. Therefore, we conducted the first large-scale analysis for revealing the actual status of cloud service abuse and discussing effective countermeasures. For this analysis, we require a large-scale observation point for various cyber-attacks that abuse cloud services. Our idea is to indirectly observe such attacks by using many different types of blacklists in combination. Our study showed some attack trends using cloud services such as attack types, regions, and anti-abuse actions.

2

Efficient Labeling for Detecting Malicious Domain Names

2.1 Indroduction

Security technologies related to the Internet along with the domain names and DNS (Domain Name System) that support it are indispensable for realizing a "Sustainable Social Information Infrastructure". While some domain names are used normally, others, known as malicious domain names, are abused by attackers. Malicious domain names are used as attack infrastructures in various cyber-attacks. According to Cisco reports, in 2015, 91.3% of cyber-attacks used a malicious domain name [1]. In addition, attackers employ sophisticated techniques to avoid general countermeasures against the use of malicious domain names. For example, they generate a large number of malicious domain names that are effective only for a short period of time using a method known as the Domain Generation Algorithm (DGA), or they reacquire benign domain names that were originally used for different purposes and abuse them [2, 3]. This suggests that there is a strong need for a technology to effectively classify malicious domain names to prevent cyber-attacks using these domain names as infrastructure.

In the situation where attackers use and throw away a large number of malicious domain names in a short period of time, the conventional countermeasure of creating a blacklist of malicious domain names has become ineffective because the size of the blacklist grows unnecessarily, and most domain names in it are not used already. Therefore, a system to classify the malicious domain names using supervised machine learning has been proposed as a new countermeasure technique. In this paper, such a system is called a malicious domain name detection system.

Many malicious domain name detection systems using supervised machine learning have been previously proposed [4, 5, 6, 2, 7, 3, 8]. Some of them are used as commercial services. For example, Notos [4] has been proposed by Manos et al. and DomainProfiler [8] has been proposed by Chiba et al. Generally, in order to apply supervised machine learning to the classification problem and obtain sufficient classification accuracy, it is necessary to prepare a large amount of data with exact class labels as training data. For example, DomainProfiler, proposed by Chiba et al., realizes its high classification accuracy by using approximately 170,000 domain names, carefully labeled as training data by specialized technicians.

Apart from paying attention to the amount of training data, it is also necessary to consider the imbalance of class labels in the training data. This is because if the labels included in the training data are extremely biased, a classifier with low classification accuracy for a minority class may be trained. In general, because there are few malicious domain names compared to benign domain names, it is necessary to take measures against such an imbalance to improve the accuracy of classifying a malicious domain name. For example, in DomainProfiler, the proportion of benign domain names to malicious domain names in the training data was adjusted to be substantially equal.

However, an examination of the labels of domain names randomly chosen from all real domain names indicates that the probability of a domain name being malicious is extremely low. Therefore, the collection of a large number of malicious domain names that can guarantee sufficient classification accuracy in a malicious domain name detection system using supervised machine learning has a large labeling cost. Moreover, most of the benign domain names collected as a by-product of collecting malicious domain names cannot be used as training data. Reducing the labeling costs under these circumstances can be said to be a common problem of malicious domain name detection systems using supervised learning. Moreover, this problem cannot be ignored in actual operations because it is necessary to continuously update the training data used to retrain the malicious domain name detection system during operation to maintain the classification accuracy for new domain names generated on a daily basis.

In this paper, we propose a labeling method based on active learning. Active learning is a machine-learning approach that implements the process of selecting the domain names to be labeled to improve the classification accuracy. Labeling only those domain names that would be useful as training data can sufficiently improve the classification accuracy of the classifier with a small amount of training data. This would also greatly reduce the cost of labeling domain names, which is defined as the number of domain names to be labeled in this paper. Accordingly, the method proposed in this paper is expected to be useful when updating a currently operating classifier by adding new training data and in the initial stages of training a new classifier when less training data are available.

Furthermore, when training a classifier with a small amount of training data, the result overly fits only those training data, with the problem that the ability to classify unknown data could not be guaranteed. In this study, we solve this problem by introducing the idea of ensemble learning, namely we take the weighted average of the prediction results of multiple classifiers.

The contribution of this paper is as follows.

- A classifier with sufficient classification accuracy can be acquired after greatly reducing the absolute number of domain names that humans need to label.
- A classifier with sufficient classification accuracy can be acquired even for domain name data in which the proportion of benign and malicious is biased.
- The classification accuracy can be maintained even under conditions where the property of the domain name continues changing.

The structure of this paper is as follows. Section 2.2 explains the application of active learning and ensemble learning proposed in this paper to a malicious domain name detection system. In Section 2.3, we describe the evaluation experiment conducted by using data of actually observed domain names, and discuss the result in Section 2.4. Section 2.5 discusses the limitation and practicality of the proposed method. Section 2.6 summarizes related work, and Section 2.7 concludes this paper.

2.2 Proposed Method

2.2.1 Motivation

As described above, improving the classification accuracy of malicious domain names by supervised machine learning generally requires two conditions to be established. In other words, (1) a large amount of data with accurate class labels must be prepared as training data and (2) the imbalance of class labels in the training data needs to be adjusted. Solving the problem described in (1) requires us to determine how to reduce the cost of labeling the domain name. This is necessary because of the excessive cost associated with labeling each domain name, for which there are two main reasons. The first is that the number of domain names continues to increase day by day regardless of whether they are benign or malicious. The second is that it is not easy to correctly determine whether the domain name is actually being used in an attack, because this would require classification by a security device and analysis by a skillful technician. Solving the problem described in (2) requires undersampling of the majority of labeled data, oversampling of the minority of labeled data, customization of the objective function in learning, or a combination thereof as a conceivable standard method. However, these countermeasures are possible when the labeled data are provided. In other words, when a label is not provided, techniques such as undersampling, oversampling, and customization of the objective function cannot be applied.

This motivated us to propose a step-by-step method to select a domain name to be labeled based on the certainty of whether revealing the label of the selected domain name would contribute to the improvement of the classification accuracy of the current classifier. By using this method, we can select a domain name to be labeled without overly biasing the benign domain name. In addition, we propose a method to integrate the classifiers trained using training data labeled at each stage. The use of these methods is expected to enable a classifier with sufficient classification accuracy and generalization ability to be acquired with a small amount of training data. In addition, this approach is expected to realize high classification accuracy regardless of the proportion of malicious and benign domain names in the training data. That is, we expect it to be possible to reduce the cost of labeling and overcome the imbalance in the proportion of class labels simultaneously. Thus, it should be noted that the issue of this paper is "Proposing the labeling method to reduce the labeling cost while maintaining classification accuracy". Our proposed method uses active learning to select domain names to be labeled, and ensemble learning for the integration of multiple classifiers.

The above idea can be applied universally to a system for classifying malicious domain names by using supervised machine learning. However, in order to improve the classification accuracy by implementing it concretely, it is necessary to adjust according to the characteristics of individual systems. Therefore, this paper does not compare the classification accuracy between multiple systems but compares multiple labeling methods for one specific system. This is because the viewpoint of evaluation in this study is the extent of reduction of labeling cost. We selected DomainProfiler proposed by Chiba et al. in this paper because it has the highest classification accuracy for malicious domain names between the existing systems. This system has the ability to classify malicious domain names with an accuracy of 99% by performing feature design focusing on the time series variation of a domain name usage situation. Another reason for choosing it is that DomainProfiler is one of the few systems whose training data, features, and algorithm have been clarified through papers. In addition to DomainProfiler, Notos [4] proposed by Manos et al. is also one of such systems. However, all features used in Notos are also used in DomainProfiler. In other words, DomainProfiler is upward compatible

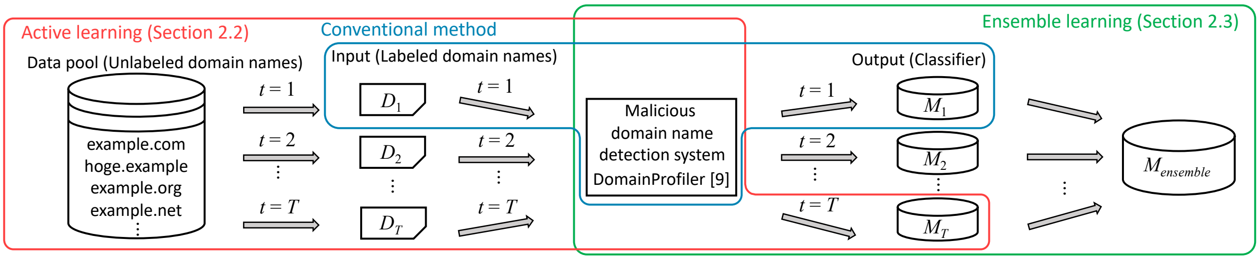

Active learning (Section 2.2)
Conventional method
Ensemble learning (Section 2.3)
Data pool (Unlabeled domain names)
example.com
hoge.example
example.org
example.net
Input (Labeled domain names)
$t = 1$
$t = 2$
$t = T$
D_1
D_2
D_T
Malicious domain name detection system DomainProfiler [9]
Output (Classifier)
M_1
M_2
M_T
$M_{ensemble}$

Fig 2.1: Relationship between proposed and conventional methods.

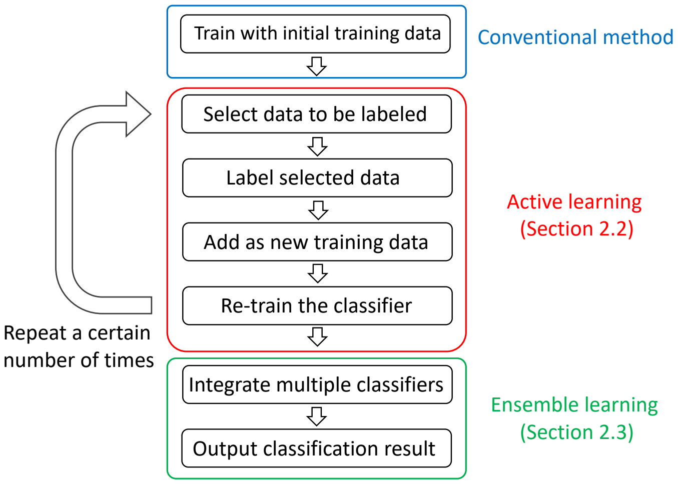

Train with initial training data
Conventional method
Select data to be labeled
Label selected data
Add as new training data
Re-train the classifier
Active learning (Section 2.2)
Repeat a certain number of times
Integrate multiple classifiers
Output classification result
Ensemble learning (Section 2.3)

Fig 2.2: Procedure of the proposed method

with Notos and the classification accuracy is greatly improved by DomainProfiler. There are many commercial malicious domain name detection systems but most of them do not reveal the technical details as described above. We consider a method to expand DomainProfiler such that it can be applied to unlabeled data. This approach reduces the labeling cost and improves classification accuracy and stability.

The relationship between the proposed method and the conventional method is illustrated in Fig. 2.1. In addition, the procedure of the proposed method is shown in Fig. 2.2. The proposed method consists of two elements: (1) a method for labeling domain names by active learning, and (2) a method for integrating multiple classifiers by ensemble learning. Further, in the conventional system typified by DomainProfiler, labeled domain name data are given as input, whereas in the proposed method domain name data without labels are used as input. In addition, the output of the conventional method is a single classifier, whereas the output of the proposed method is a model integrating multiple classifiers. The proposed method therefore extends and improves the conventional method. We explain the application of active learning (1) in Section 2.2.2, and the application of ensemble learning (2) in Section 2.2.3.

2.2.2 Selection of Training Data to be Labeled by Active Learning

Our proposed method introduces the idea of active learning, which is used as the domain name labeling method. Through active learning, a series of processes is performed in which unlabeled domain names are selected to be labeled. They are then labeled, added as new training data, and the classifier is retrained. Active learning entails selecting only data considered to contribute to the improvement of the classification accuracy of the classifier at that time in stages and labeling these data. Therefore, it is possible to improve the classification accuracy of the classifier while suppressing the cost of acquiring the labeled training data.

Many methods for selecting data to be labeled have been proposed, depending on the method for evaluating the degree to which it is expected to contribute to improving the classification accuracy of the classifier [9]. Among them, *uncertainty sampling* is the simplest and most commonly used method. Uncertainty sampling includes *margin sampling* that selects the data with the smallest difference between the largest class membership probability and the next largest, *least confident*, which selects the data of which the largest class membership probability is the smallest, and *entropy-based sampling* that selects the data of which the entropy of the class membership probability is the largest. Because these methods are all equivalent if they are used in two-class classification problems, we used margin sampling for preliminary verification in our study. The preliminary verification results confirmed that margin sampling is performed fast and is also able to effectively solve the problem of domain name classification with which this research is concerned. Therefore, margin sampling was adopted in our study.

Fig. 2.3 illustrates the concept of active learning using margin sampling. This figure explains the way in which the decision boundary of classification of the classifier is updated by margin sampling. The points depicted in the figure represent data, here comprising both labeled and unlabeled data. In active learning using margin sampling, when classifying with the most recently trained classifier (hereinafter referred to as the “Latest Classifier”), data located in the vicinity of the decision boundary (i.e., gray area) of classification are selected and labeled by an oracle (The term “oracle” refers to an expert or annotator). This method is based on the concept that, if the label of the data that cannot be reliably classified by the Latest Classifier is known, the possibility of improving the classification accuracy of the classifier is high. This concept is the most important foundation on which to determine which domain name to label in a situation in which the label is not known. If there is no dedicated labeling method, a domain name must be selected to be labeled randomly. The labels of randomly selected domain names will be extremely biased to benign because the proportion of malicious domain names in the actual domain names is very small. In general, such biased domain

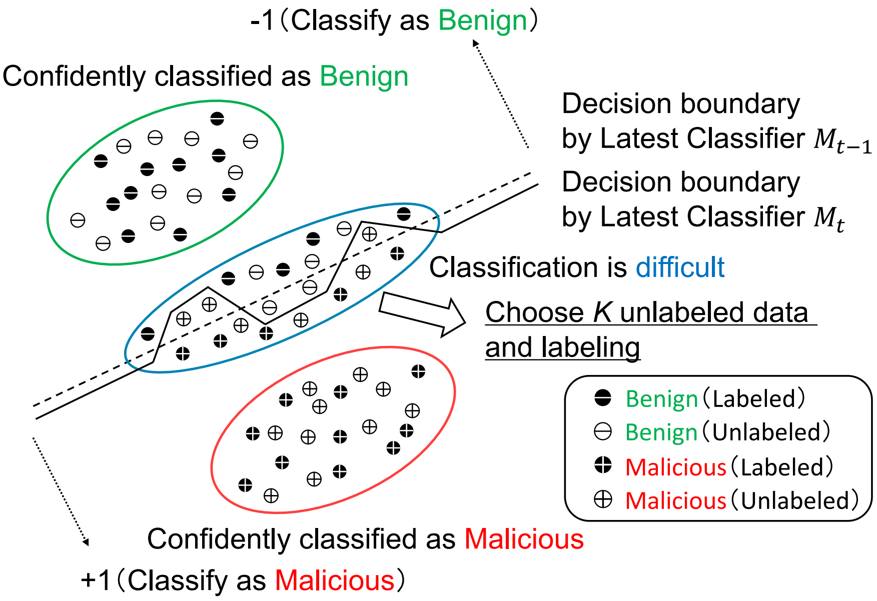

-1（Classify as Benign）
Confidently classified as Benign
Decision boundary by Latest Classifier M_{t-1}
Decision boundary by Latest Classifier M_t
Classification is difficult
Choose K unlabeled data and labeling
● Benign（Labeled）
⊖ Benign（Unlabeled）
⊕ Malicious（Labeled）
⊕ Malicious（Unlabeled）
Confidently classified as Malicious
+1（Classify as Malicious）

Fig 2.3: Conceptual diagram of active learning.

names cannot contribute to the improvement of accuracy of the classifier. Consequently, the labeling cost is wasted. In contrast, active learning enables us to select the domain name that can contribute to the improvement of accuracy of the classifier, thereby labeling a sufficient number of malicious domain names. The specific procedure of margin sampling is as follows.

Labels to be provided to malicious and benign domain names are $\{+1, -1\}$, respectively. The Latest Classifier M predicts the label, $y_i \in \{+1, -1\}$, of the unlabeled domain name, $x_i \in D$. At this time, the probability that x_i is predicted to be malicious by the Latest Classifier M is defined as $P_{i,+1}$ and the probability that x_i is predicted to be benign is defined as $P_{i,-1}$. Then, among domain names in D, K domain names are selected in ascending order of the difference between the probability of being predicted to be malicious and the probability of being predicted to be benign, $|P_{i,+1} - P_{i,-1}|$. Finally, we label the selected K domain names, add it as new training data, and retrain the classifier. Through a series of flows, the decision boundary of the Latest Classifier is updated from the dotted line to the solid line in Fig. 2.3.

2.2.3 Integration of Classifiers by Ensemble Learning

In the method described in Section 2.2.2, classification is performed using only the Latest Classifier. At this time, because the Latest Classifier is trained with a small amount of training data, the classification ability for unknown data is not guaranteed and this becomes a problem. This led us to propose a method to integrate all the classifiers that are trained sequentially by active learning each time training data are added, by introducing the concept of ensemble learning. Reflection of the prediction results of previously trained classifiers in the prediction result of the Latest Classifier is expected to enable a prediction result with higher generalization ability to be obtained as output. Typical methods based on ensemble learning include Random Forest [10], Bagging [11] and Boosting [12], and many other methods derived from these methods have been proposed. Below, we explain the application of ensemble learning to this work.

Assume that T classifiers have been trained thus far. Let M_t be the classifier created for the t th time and N_t the number of training data used to train M_t. Here, $N_t = N_{t-1} + K$. Further, define the prediction function by M_t as $h_t(x) \to \{+1, -1\}$. In this work, we integrate the output of $h_t(x)$ based on a weighted average, where the weight, α_t, is set to be proportional to N_t, as follows.

$$\alpha_t = \frac{N_t}{\sum_{t=1}^{T} N_t} \tag{2.1}$$

Considering the fact that the characteristics of domain names continue changing on a daily basis, it is reasonable to focus on the prediction results of classifiers trained using training data that contain many newly generated domain names. The weight defined in (2.1) is designed by taking this into consideration.

Finally, the final prediction function $H(x)$ is defined as follows, and the label y of data x of a certain domain name is predicted as:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \quad y = \begin{cases} +1 & (H(x) \geq 0) \\ -1 & (H(x) < 0) \end{cases} \tag{2.2}$$

2.3 Experimental Setup

2.3.1 Domain Name Data to Use

We verified the effectiveness of the proposed method by creating a data set of domain names by designing a feature based on the same definition as DomainProfiler by Chiba et al. [8]. The features used in this study can be divided into three types. The first is obtained from the time-series change of the domain name, the second is obtained from the IP address of the domain name, and the third is obtained from the string of the domain name. The specific definition of the features is presented in Table 2 of the paper on DomainProfiler [8]. The number of features belonging to each type and the corresponding type names used in the paper on DomainProfiler are listed in Table 2.1 in this paper. However, it should be noted that this study uses some additional features as well. For example, this study uses the various

Table 2.1: Details of features.

Type	Time series change	IP address	Domain names
Number of features	80	18	17
Corresponding names	TVP	rIP	rDomain

observation periods of the feature in DomainProfiler: "whether it was on the public blacklist during the observation period". The data set contains both malicious and benign domain names. As the malicious domain names, as of May 29, 2018, we used those posted on the public blacklist hpHosts [13]. This blacklist includes malicious domain names confirmed to be used as attack sites, malware distribution sites, and phishing sites. On the other hand, as the benign domain names, as of May 29, 2018, we used the top 100,000 domain names listed on Alexa Top Sites [14], which provides a list of web sites ranked by traffic volume. However, Alexa Top Sites only contains the second-level domain (e.g., `example.com`) and often does not provide the IP address information corresponding to the domain name, which is a necessary feature of DomainProfiler. Therefore, we used the search engine to extract and use the FQDN, which has the IP address under the second-level domain (e.g., `www.example.com`). To as much as possible eliminate the possibility of mistakenly including a benign domain name in the set of malicious domain names and vice versa, at the time of evaluation, we adopted only domain names verified by VirusTotal [15] and multiple commercial security data.

Hereinafter, the set of domain names that can be added as training data is referred to as the "data pool". In this study, we evaluate the versatility of the proposed method by conducting experiments not only on the data pool that are adjusted beforehand such that the proportion of class labels is not biased, but also on the data pool that contains a more realistic proportion of class labels. To this end, we prepared two types of data pools, the first with a sufficient proportion of malicious domain names and the other with extremely few malicious domain names in proportion. The data sets with the former and the latter as their data pools are termed data pool I and data pool II, respectively. Table 2.2 shows the number of training data for each label, feature dimension, and creation date of each data set. In the evaluation experiments using both data pools I and II, a total of 20,000 domain names including 10,000 benign and malicious domain names are used as test data. These 20,000 domain names are common regardless of whether we use data pool I or data pool II for the evaluation experiments.

The domain name to be added as training data is always selected from the domain name in the data pool, and once selected the domain name is excluded from the data pool. Furthermore,

Table 2.2: Details of the created data set.

	Data pool I	Data pool II
Number of malicious training data	125,720	815
Number of benign training data	81,563	81,563
Total number of data pools	207,283	82,378
Feature dimension	115	115
Creation date	May 29, 2018	May 29, 2018

originally, the domain name included in the data pool is not labeled, and it is not known whether it is benign or malicious. However, in this study, all domain names are labeled in advance of creating the data set. Therefore, we hide the label of domain names included in the data pool, and process all domain names as though their labels are unknown. Our proposed method regards restoring hidden true labels as a new labeling operation in our evaluation experiments. In actual operations, it is necessary for an oracle to label a domain name that has been earmarked for addition as training data.

2.3.2 Selection of Machine-Learning Algorithm

We selected to use Random Forest as a machine-learning algorithm to train a classifier using given training data. Random Forest is also the machine-learning algorithm that was adopted for DomainProfiler. Random Forest, which determines the final prediction result by integrating the prediction results of multiple decision trees trained using randomly selected data from the training data set, is based on ensemble learning, as described in Section 2.2.3. Thus, the proposed method uses Random Forest to train the classifier using the newly labeled training data, and subsequently uses ensemble learning when integrating the previously created classifiers. It is also possible to use machine-learning algorithms other than Random Forest to train the classifier.

2.3.3 Metric and Viewpoint of Evaluation

We use Accuracy, as defined by (2.3), as a metric to evaluate the classification accuracy of the classifier. Here, in (2.3), true positive (TP) is the number of times the malicious domain name is correctly predicted as malicious, true negative (TN) is the number of times the benign domain name is correctly predicted as benign, false negative (FN) is the number of times a malicious domain name is erroneously predicted as benign, and false positive (FP) is the number of times a benign domain name is erroneously predicted as malicious.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.3}$$

Accuracy is an index to judge the degree to which the classification result corresponds to the true label. Accuracy takes a value between 0 and 1, and as the value approaches 1, it indicates that the classification accuracy of classifier is high. The value of Accuracy is an incomplete indication when the labels of the test data are biased. For example, if all the test data consist of malicious domain names, the value of Accuracy will be 1.0 when any domain name is simply classified as malicious. Therefore, the test data set used in the evaluation experiment was constructed such that it included the same number of benign and malicious domain names. This ensures that Accuracy is a reasonable metric for evaluating the classification accuracy in our evaluation experiments.

The effectiveness of the proposed method was verified by conducting an evaluation experiment based on the following viewpoint. The method we selected with which to compare and evaluate the active learning method randomly selected domain names from the data pool to be added as training data. Hereafter this method is referred to as random sampling.

- Comparison of the Accuracy for selecting training data to add when using random sampling or active learning.
- Comparison of the Accuracy when only the Latest Classifier's prediction results are used and when we integrate the prediction results of past classifiers by using ensemble learning.

2.3.4 Setting the Experimental Parameters

The number of domain names that were randomly extracted from the data pool for use as initial training data is 500. The number of training data to be added at a time is 25, and the classifier is iteratively trained until the total number of training data reaches 5,000 in data pool I and 2,500 in data pool II. We trained the classifier using Random Forest with 10 decision trees. Every time the classifier is trained, the performance is evaluated by using a test data set containing 10,000 malicious and benign domain names. The number of initial training data, the number of training data to be added at once, and the total number of training data were determined as being suitable for verifying the effectiveness of the proposed method based on the results of the preliminary verification. Note that the set of test data is fixed with the above

Table 2.3: Details of experimental parameters.

	Data pool I	Data pool II
Number of initial training data	500	500
Number of training data added at once	25	25
Total number of training data	**5,000**	**2,500**
Number of malicious test data	10,000	10,000
Number of benign test data	10,000	10,000

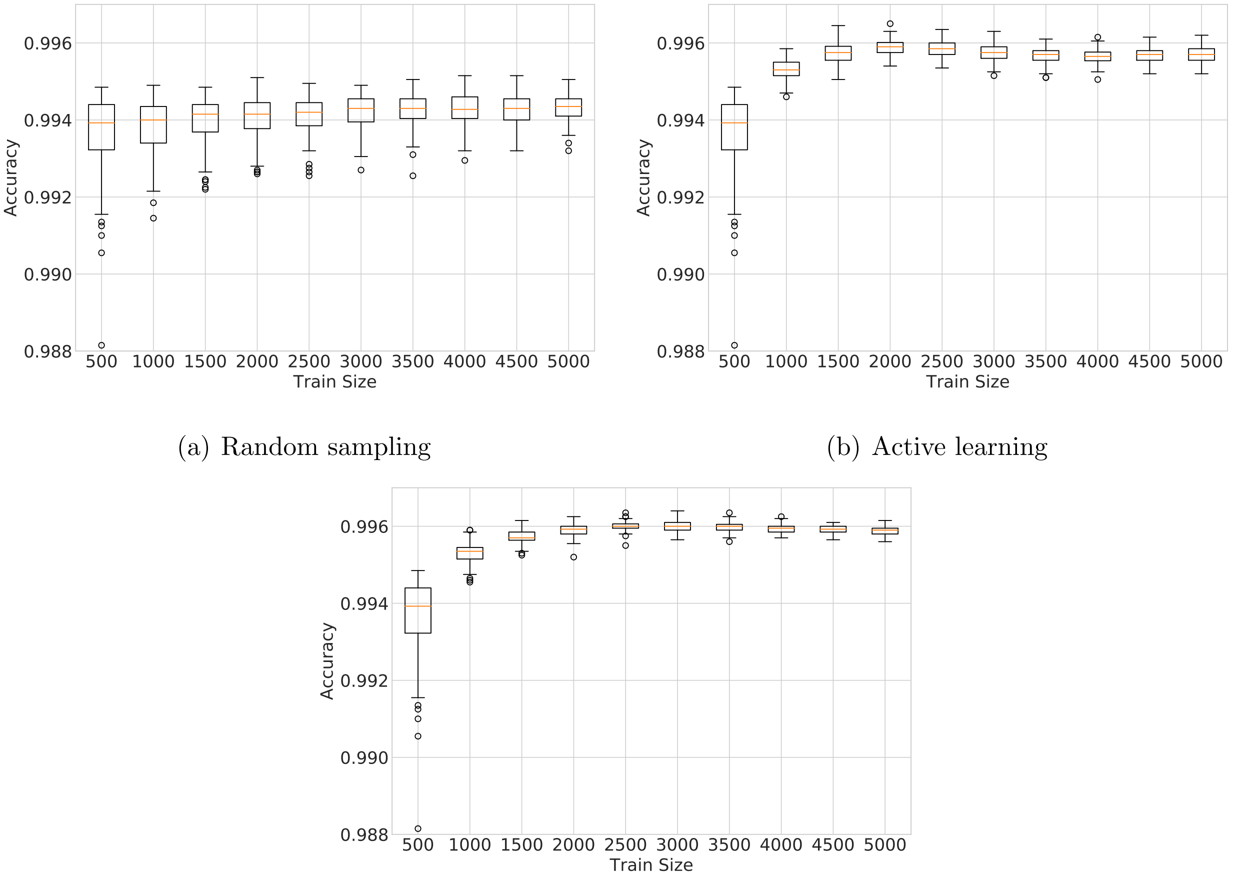
20,000 domain names, regardless of the data pool used for training. The above parameters are summarized in Table 2.3.

Active learning is greatly affected by the initial classifier trained by the initial training data, and the domain names selected as training data are expected to change. Therefore, the proposed method was applied to 100 kinds of initial training data that were created by changing the seed value of the random number in random sampling. That is, for each data set, the proposed method is executed 100 times during which the fluctuation of the execution result is observed. This makes it possible to evaluate the stability of the classifier trained by the proposed method in terms of the fluctuation of the classification accuracy.

2.4 Experimental Results

The proposed method was evaluated based on the settings in Section 2.3. The structure of this section is as follows. Section 2.4.1 presents the evaluation of the effectiveness of the proposed method. Specifically, we describe the effectiveness of active learning in Section 2.4.1, the effectiveness of ensemble learning in Section 2.4.1, and evaluate the contribution of the proposed method in Section 2.4.1.

In Section 2.4.2, we analyze the behavior of the proposed method. We consider the reason why active learning was effective in Section 2.4.2. In addition, we discuss variations in the classification results when focusing on individual domain names in Section 2.4.2.



(a) Random sampling

(b) Active learning

(c) Active learning + Ensemble learning

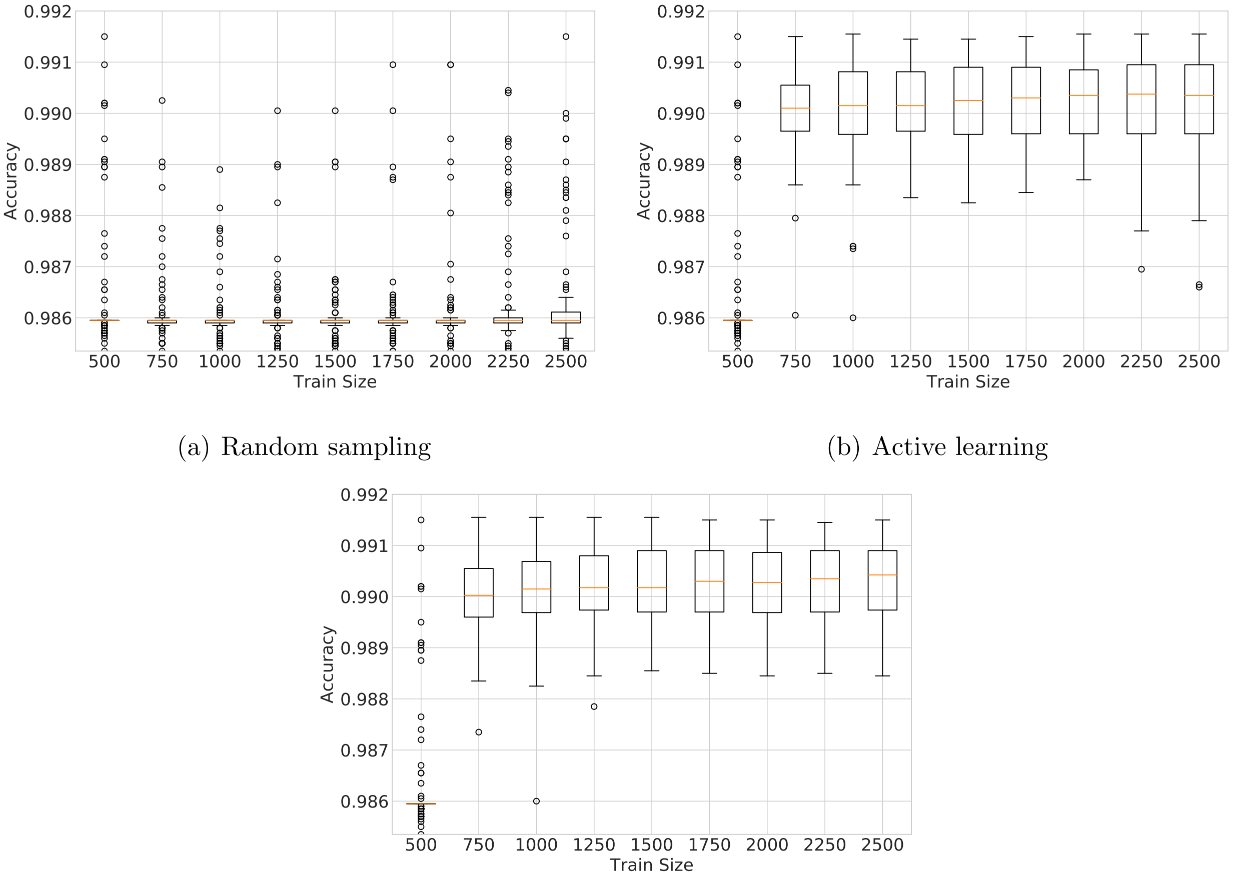
Fig 2.4: Variation of Accuracy when each method is applied (data pool I)

2.4.1 Performance Evaluation

Effectiveness of Active Learning

We conducted experiments to compare and assess the effectiveness of the two methods, random sampling and active learning, for selecting the training data to be added. Figs. 2.4 and 2.5 show the results of the experiments using data pools I and II, respectively. The vertical and horizontal axes show the Accuracy and the total number of training data used for training the classifier, respectively. Incidentally, in each of the following figures including these figures, the scattering of the execution results for 100 kinds of initial training data is represented by the quantile of the box-and-whisker plot.

Figs. 2.4(a) and 2.5(a) show that, even if the training data used for training the classifier are added by the random sampling method, the value of Accuracy in the case of data pool I remains approximately 0.994, and in the case of data pool II it remains approximately 0.986, indicating that the classification accuracy of the classifier does not improve. Data pool II has



(a) Random sampling

(b) Active learning

(c) Active learning + Ensemble learning

Fig 2.5: Variation of Accuracy when each method is applied (data pool II)

Table 2.4: Breakdown of benign and malicious domain names added to the set of training data.

Applied method	Benign number	Malicious number
Random sampling (data pool I)	1,768	2,732
Active learning (data pool I)	1,541	2,959
Random sampling (data pool II)	1,985	15
Active learning (data pool II)	1,893	107

lower classification accuracy than data pool I. The reason is the consequence of the proportion of malicious domain names in training data remaining low even if the total number of training data increases.

Table 2.4 presents the breakdown of benign and malicious domain names when 4,500 training data are added from data pool I and 2,000 training data are added from data pool II. The results in this table show that, when training data are added from data pool II using random sampling, only 15 malicious domain names are included in the set containing 2,000 training data. Therefore, when the random sampling method is used to randomly select a domain name to be added as training data, it is impossible to appropriately select data that contribute to improving the classification accuracy of the Latest Classifier; hence, high-cost labeling does not improve the classification accuracy.

On the other hand, as shown in Figs. 2.4(b) and 2.5(b), when the active learning method is used to select domain names to be added as training data, the classification accuracy is improved as the number of training data increases and converges to a nearly constant value. The results in Table 2.4 confirm that the use of active learning on both data pools I and II leads to the addition of a greater number of malicious domain names as training data than the original proportion in the data pool. In fact, the number of malicious domain names that were added from data pool II was approximately seven times more than when using random sampling. That is, the number of malicious domain names selected by the proposed method has increased from 15 to 107. Here, it is important to focus on the fact that the classification accuracy of the classifier improves when trained with the domain names selected by our proposed method. This indicates that the degree of increase in the number of malicious domain names selected by our proposed method is effective for improving the classification accuracy. Figs. 2.4(b) and 2.5(b) confirm that the domain name selected by our proposed method was effective in improving the classification accuracy. This is an effective approach to select domain names to be added to the training data, because it is based on criteria that enable us to evaluate whether revealing the label of the domain name can contribute to improve the classification accuracy of the current classifier.

Next, the number of training data required to improve the classification accuracy of the classifier was investigated for random sampling. Fig. 2.6 shows the variation in the value of Accuracy when random sampling reaches 200,000 domain names of training data on data pool I. In addition, Fig. 2.7 shows the variation in the value of Accuracy when random sampling reaches 80,000 domain names of training data on data pool II. We compare Figs. 2.4(b) and 2.6, and Figs. 2.5(b) and 2.7, respectively. The classification accuracy of the classifier trained by using 200,000 and 80,000 domain names randomly selected from data pool I and II, respectively, can be equaled by using the classifier trained using active learning with only approximately 2,000 domain names. The above results show that appropriately selecting training data by active learning enables the classification accuracy of the Latest Classifier to be improved with a small amount of training data regardless of the proportion of benign and malicious domain names in the data pool. Thus, the above results indicates that the issue

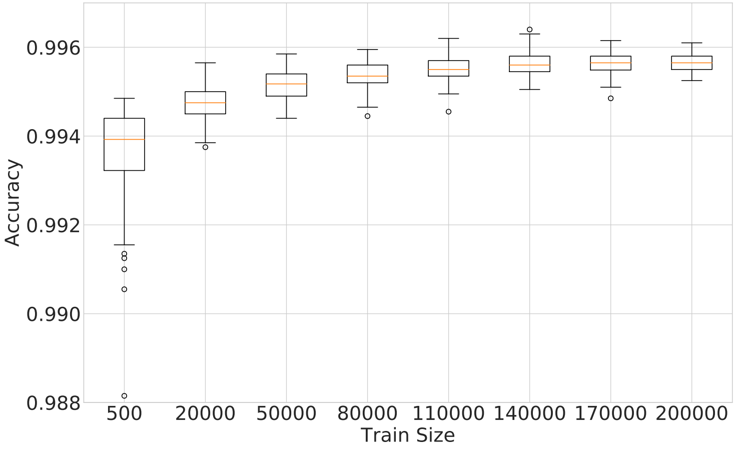


Fig 2.6: Variation of Accuracy by random sampling (data pool I)

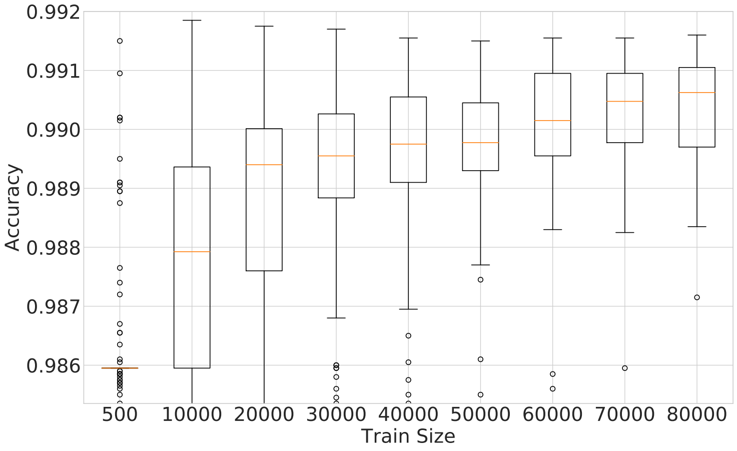
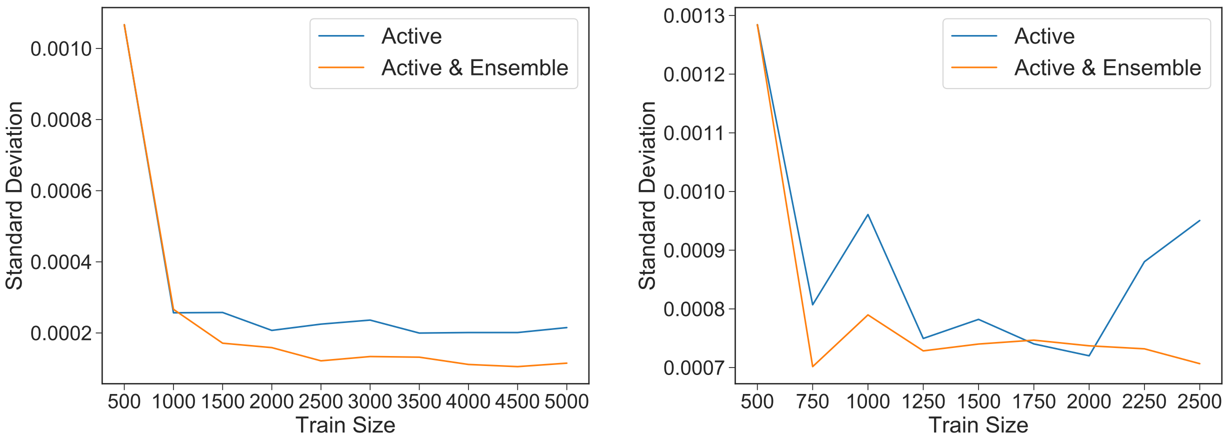


Fig 2.7: Variation of Accuracy by random sampling (data pool II)

of this paper: "Proposing the labeling method to reduce the labeling cost while maintaining classification accuracy" has been achieved by the proposed method.

Effectiveness of Ensemble Learning

We also evaluated the extent to which the use of ensemble learning can improve the generalization ability of the prediction results. This was achieved by comparing the prediction results obtained by using only the Latest Classifier with those obtained when integrating multiple classifiers using ensemble learning. In the results shown in Figs. 2.4(a), 2.4(b), 2.5(a), and 2.5(b), each time the classifier is retrained, the previous classifier was discarded and only the prediction of the Latest Classifier was used. On the other hand, for the results shown in Figs. 2.4(c) and 2.5(c), rather than discarding the previous classifier trained by active learning,



(a) Data pool I

(b) Data pool II

Fig 2.8: Variation of standard deviation of Accuracy

it was integrated with the Latest Classifier using the ensemble learning method. These results were obtained by the full set of methods (conventional, active learning, and ensemble learning) shown in Fig. 2.1. By comparing Figs. 2.4(b) and 2.4(c), Figs. 2.5(b) and 2.5(c), respectively, by using ensemble learning, when the number of training data in data pool I is 2,000 or more, the median of Accuracy stabilizes at approximately 0.996, and when the number of training data in data pool II is 1,500 or more, Accuracy stabilizes at 0.988 or more. For example, when the active learning method was used, the classification accuracy sometimes decreased or the fluctuation increased despite the addition of training data (Fig. 2.4(b): 3,000 to 3,500, Fig. 2.5(b): 2,000 to 2,250). In addition, for each experiment using data pool I and data pool II, we show the graph with the standard deviation of Accuracy calculated using a total of 100 experiments on the vertical axis and the number of training data on the horizontal axis in Fig. 2.8. Fig. 2.8 compares the results of using only active learning and combining active learning and ensemble learning (proposed method). From Fig.2.8, we can confirm that the above problem of fluctuation was overcome by using ensemble learning. This means that the generalization ability can be improved by using ensemble learning even when training the classifier with a small amount of training data.

Improvement of Classification Accuracy

Based on the above discussion, it was found that the classification accuracy can be improved by combining active learning and ensemble learning, regardless of the proportion of class labels in the data pool. This was confirmed by calculating the statistical information for the values

Table 2.5: Accuracy for 5,000 training data (data pool I)

Applied method	Minimum	Median	Maximum	Standard deviation
Random	0.9932	0.9944	0.9951	0.000339
Active	0.9952	0.9957	0.9962	0.000215
Active & Ensemble	0.9956	0.9960	0.9962	0.000115

Table 2.6: Accuracy for 2,500 training data (data pool II)

Applied method	Minimum	Median	Maximum	Standard deviation
Random	0.9850	0.9860	0.9915	0.001215
Active	0.9866	0.9904	0.9916	0.000951
Active & Ensemble	0.9885	0.9904	0.9915	0.000707

Table 2.7: Rate of change compared to random sampling [%].

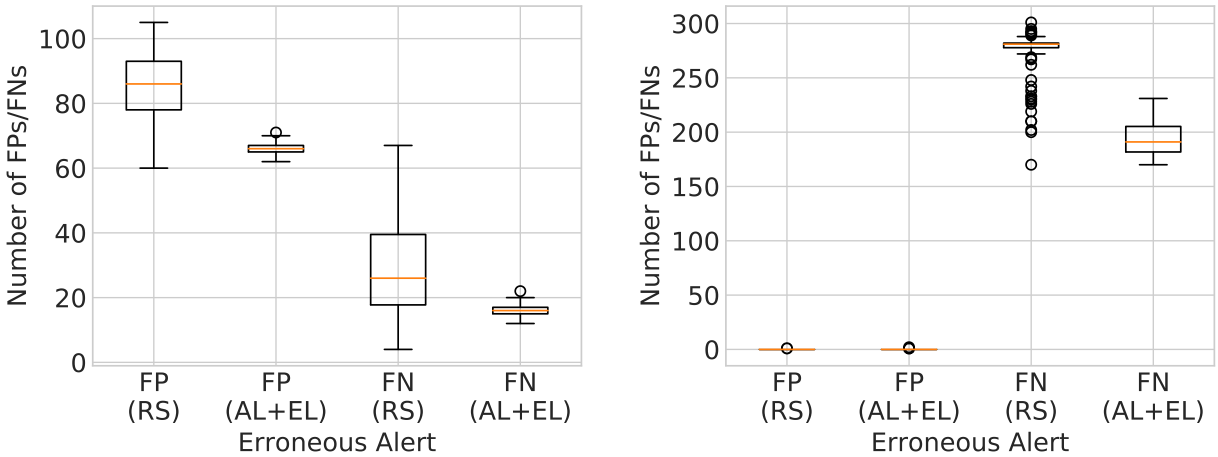
Applied method	Minimum	Median	Maximum	Standard deviation
Active (I)	0.20	0.13	0.11	-37
Active & Ensemble (I)	0.24	0.16	0.11	-66
Active (II)	0.16	0.45	0.01	-22
Active & Ensemble (II)	0.36	0.45	0.00	-42

of Accuracy when each method was applied to data pool I and data pool II, respectively. The results are provided in Table 2.5, Table 2.6, and Table 2.7. The results in Table 2.5 indicate that, by using active learning and ensemble learning on data pool I, the median of Accuracy when the number of training data is 5,000 is increased by approximately 0.16% compared to the use of random sampling only. This number is a very small and seemingly low value. However, the median of Accuracy for random sampling is 0.9944, meaning that improving 0.0016 out of the remaining $1 - 0.9944 = 0.0056$ is significant. Additionally, Table 2.7 shows the improvement of the minimum, median, maximum, and standard deviation of Accuracy of the proposed method in comparison to those of random sampling when the number of training data is 5,000 in data pool I and 2,500 in data pool II. In Table 2.7, data pool I is abbreviated as I and data pool II is abbreviated as II. Table 2.5 also shows that the minimum value of Accuracy increases by approximately 0.24% and the standard deviation of Accuracy decreases by approximately 66% when the number of training data is 5,000 in data pool I. The results in Table 2.6 confirms that the minimum value of Accuracy increases by approximately 0.36% and the standard deviation of Accuracy decreases by approximately 42% when the number of data is 2,500 in data pool II. In this way, the proposed method combining active learning and ensemble learning enables us to acquire a classifier with classification accuracy equal to or higher than that of the classifier trained by the conventional method by using a small amount of training data regardless of the data pool used for training.

This seemingly small numerical improvement in Accuracy is very important in actual operation. VERISIGN shows that more than 300 million domain names were registered in the first quarter of 2019. It also shows that the number of registered domain names has increased by 3.1 million (0.9%) in comparison to the fourth quarter of 2018 [16]. Thus, it is evident that a very large number of domain names are already registered, and this number is increasing year by year. In addition, the malicious/benign classification of a domain name must be repeated many times because the use of domain names changes over time, and the people who use them change owing to expiration or re-registration. Therefore, even if the domain name is the same, the result of classification at a certain point is not always correct in the future and cannot be used continuously. In other words, for example, if the number of new domain names increases by 3.1 million, it is necessary to classify 3.1 million in addition to all the previously classified domain names (not just 3.1 million). Thus, because the number of domain names classified by commercial security services becomes enormous, the number of correctly classified malicious domain names also greatly increases even if the improvement of classification accuracy is small.

Moreover, organizations are known to spend an average of 395 hours per week to address the problems associated with misclassification of malware infections, costing approximately $1.2 million per year [17]. Similarly, to effectively respond to security incidents, spending valuable resources such as skilled human resources and budgets on false positive response is highly inefficient [18]. In other words, a system designed to detect malicious domains, such as that described in this paper, would be beneficial even if it were able to reduce misclassification by one case. Considering that 20,000 domain names were used as the test data this time, the increase of 0.16% in the median of Accuracy in data pool I, for example, corresponds to a decrease in the number of false detections of approximately 32.

In addition, Figs. 2.9(a) and 2.9(b) show the variation in the number of FPs and FNs for the method that uses only random sampling (RS in the figure) and the method that uses both active learning and ensemble learning (AL+EL in the figure). The number of training data used to produce this figure is 5,000 in data pool I and 2,500 in data pool II. The results in Figs. 2.9(a) and 2.9(b) confirm that both FP and FN for data pool I and FN for data pool II are improved by combining active learning and ensemble learning, respectively. Under actual



(a) Data pool I

(b) Data pool II

Fig 2.9: Difference in the number of FPs and FNs.

operating conditions, because more domain names are classified by the system, the number of misclassifications increases proportionally. This suggests that the contribution to actual operations by improving Accuracy realized by the proposed method is extremely large.

Here we consider the reason why the number of FPs for data pool II is almost 0 regardless of the method. In data pool II, the proportion of malicious labels to benign labels in the training data is extremely small. Therefore, training a benign domain name is considered to advance more than training a malicious domain name, and the trained classifier is more likely to classify a domain name as being benign than to classify a domain name as being malicious. Even though FP is almost 0, not all domain names are classified as benign. Indeed, in data pool II, the mean of 100 experimental values of TP for 2,500 training data is approximately 9,720 in the case of random sampling, whereas when active learning and ensemble learning are combined it is approximately 9,810. Thus, malicious domain names were successfully detected.

In addition, we consider the reason why the proposed method is more effective in data pool II than in data pool I. In data pool II, the ratio of malicious domain names in the data pool is very low. Therefore, random sampling cannot sample enough malicious domain names. In contrast, active learning preferentially samples the domain names that can contribute to the improvement of classification accuracy of the classifier. Consequently, the malicious domain names can be selectively sampled irrespective of their ratio in the data pool, and the classification accuracy of the classifier can be improved with a small amount of training data. Therefore, the proposed method is more effective in the case of data pool II where the number of malicious domain names is significantly smaller than that of benign domain names, which is an advantage of the proposed method.

2.4.2 Detailed Analysis of the Proposed Method

Analysis of Training Data Added by Active Learning

In this section, we discuss the reasons as to why active learning is effective in selecting domain names to be labeled and added to training data. Here, we suppose that the state in which the domain name is included in the training data, has various features, is ideal for training the classifier with high classification accuracy. In this state, the features of domain names in the training data are not similar to each other. Therefore, we analyzed the similarity between domain names added to the training data in detail. The similarity between domain names is analyzed by clustering. We investigate the similarity of each of the 4,500 domain names added by active learning and those added by random sampling.

We use DBSCAN [19] as a clustering method. DBSCAN is a density-based clustering method, which does not assume that clusters are spherical and does not determine the number of clusters in advance. Furthermore, not all data belongs to any particular cluster, and data that does not form a cluster is regarded as a noise point. Therefore, by using DBSCAN clustering, if there are many domain names forming a cluster, it can be assumed that there are multiple similar domain names added as training data, and if there are few domains forming cluster, then there are multiple non-similar domain names. In this paper, the feature used in DBSCAN is the 115-dimensional feature shown in Table 2.2, and Euclidean distance is used for distance calculation in clustering.

Fig. 2.10 shows a conceptual diagram of clustering domain names added as training data by DBSCAN. In Fig. 2.10, N clusters are formed, and domain names not included in any cluster are shown as noise points. Here, domain names included in the same cluster have similar features. On the other hand, domain names do not have similar features if they are not in the same cluster but in different clusters or at different noise points. Furthermore, the numbers in parentheses attached to the domain name indicates the number of rounds after which the domain name is added to the training data. Hence, domain names with the same number in parentheses are simultaneously added to training data, and domain names with different numbers in the parentheses are added to training data with different rounds.

Table 2.8 shows the results of DBSACN clustering of each of the 4,500 domain names added to training data, by two methods. Each value shown in Table 2.8 is the average of 100 experiments using different initial training data. According to Table 2.8, the number

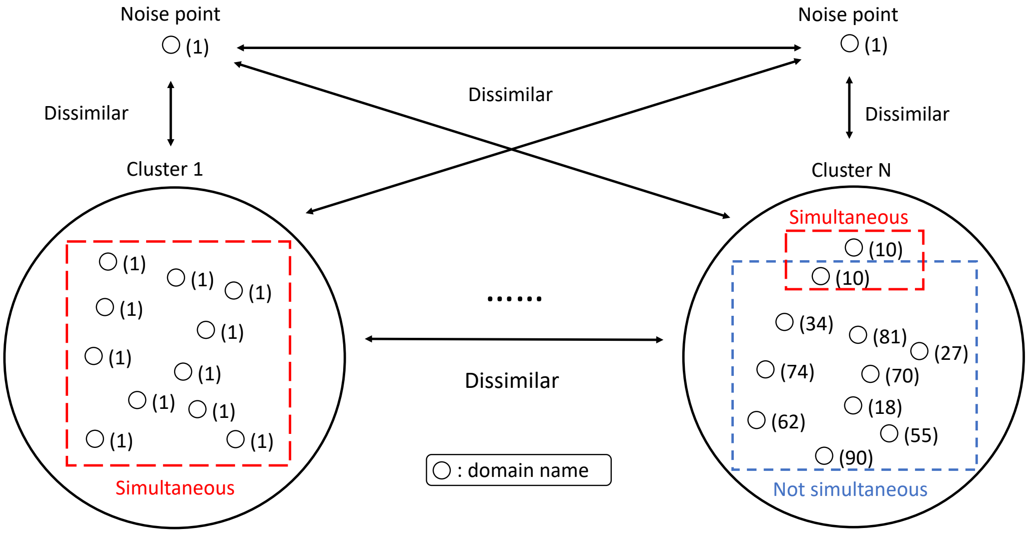


Fig 2.10: Overview of clustering domain names.

Table 2.8: Analysis of domain names added as training data (data pool I)

	Random Sampling	Active Learning
Domain names forming clusters	1,224.82	125.90
Formed clusters	90.30	16.83
Domain names per cluster (Avg.)	13.60	7.46
Difference additional rounds (Avg.)	13.16	1.89

of clusters formed by DBSCAN and the number of domain names forming clusters among the 4,500 domain names are significantly smaller when using active learning than when using random sampling. This means that the features of domain names added by active learning are less similar compared to those when using random sampling. Consequently, in the case of active learning, domain names with various features became training data, and a classifier with high classification accuracy could be learned.

Next, we focus on the timing when the domain name included in each cluster is added to the training data. As proposed in this paper, in active learning, the process of labeling K domain names selected from the data pool, adding to the training data and training a classifier is repeated. In the experiments using data pool I, the number of initial training data is 500, the number of training data added at one time is 25 ($= K$), and the addition of training data is repeated until the total number of training data reaches 5,000. Therefore, the addition of training data will be performed for 180 ($5,000 = 500 + 25 \times 180$) rounds. Here, we examine the difference in the number of rounds in which domain names were added to training data.

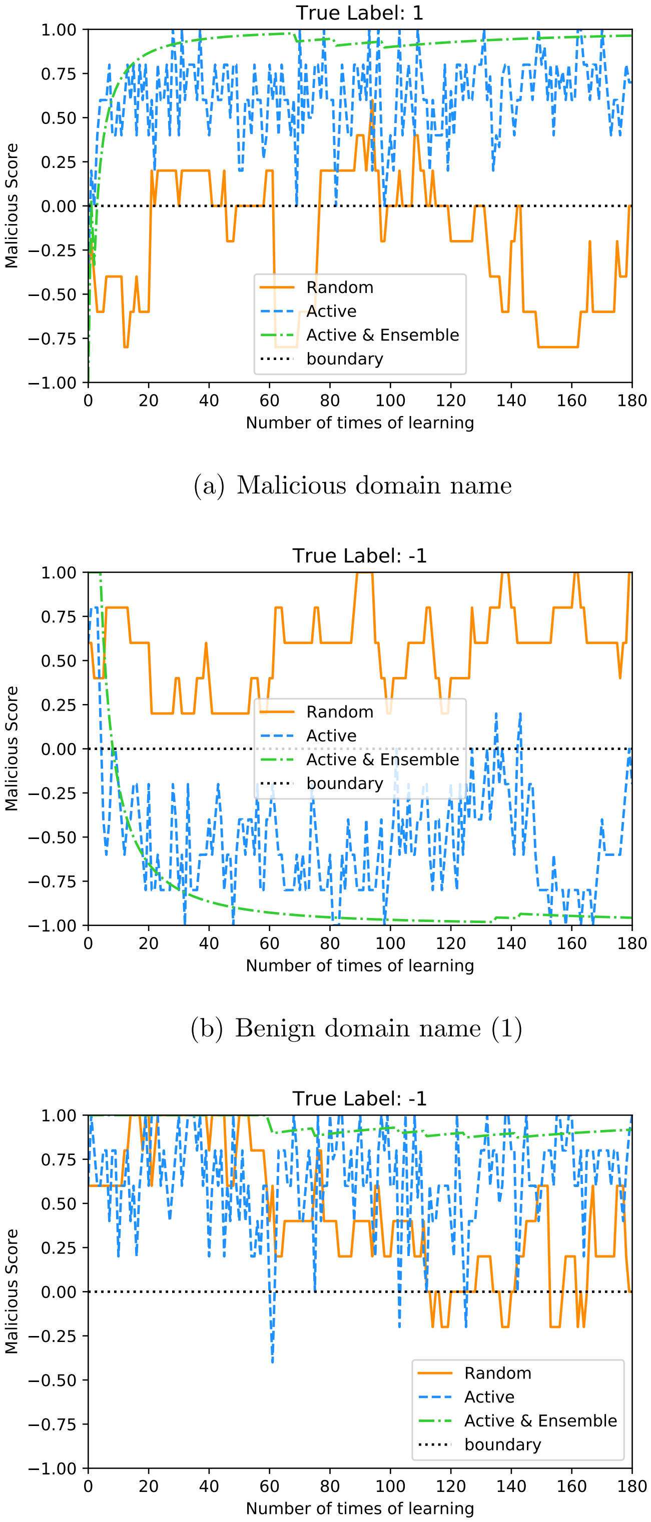
Table 2.8 shows the average number of domain names included in one cluster and the average difference of additional rounds of domain names included in one cluster. If these values are equal, it can be said that all domain names included in the cluster are added to training data in different rounds. According to Table 2.8, in the case of random sampling, the average number of domain names and the average difference of additional rounds are almost equal, but in the case of active learning, the average difference of additional rounds is significantly less than the average number of domain names. This means that in random sampling, domain names that are similar to domain names added in past rounds are added again in subsequent rounds, but in active learning, domain names that are similar to domain names added in past rounds, are not to be added again later. Hence, in active learning, domain names which have features that are not similar to those of domain names in the training data at that time are selected and newly added to the training data. This behavior is similar to what we expected because it leads to the addition of domain names with various features in the training data.

To summarize, when using active learning, domain names with various features are added to the training data, and domain names with similar features are not added in different rounds. We consider that such characteristics of active learning are factors that can improve the classification accuracy of training models by using a small amount of training data.

Time Series Variation of Individual Domain Names

Up to this point, the classification accuracy of the classifier for the entire set of test data has been quantitatively evaluated using Accuracy. In this section, we focus on each domain name in the test data and analyze how the classification result for the domain name varies with the addition of training data. The domain name to be analyzed is one that was misclassified at least once among 20,000 test data by using random sampling, active learning, and the method combining active learning and ensemble learning. The number of the misclassified domain names was approximately 320 on average of the experiments carried out by changing 100 initial training data.

In order to characterize the classification result, we define "Malicious Score" as a metric to evaluate the confidence of the classification of the target domain name. In random sampling and active learning, the Malicious Score is defined as $(P_{i,+1} - 0.5) \div 2$ based on the probability $P_{i,+1}$ that an unlabeled domain name is predicted as malicious by a classifier. For ensemble learning, $H(x)$ in (2.2) is defined as the Malicious Score. This score indicates that the target domain name is malicious when the score is 0 or more, and benign when the score is smaller than 0.



(a) Malicious domain name

(b) Benign domain name (1)

(c) Benign domain name (2)

Fig 2.11: Variation of Malicious Score of certain domain name in the test data.

Figs. 2.11(a) to 2.11(c) visualize the changes in the Malicious Score of a certain domain name in the test data. The horizontal and vertical axes represent the additional round of training data and the Malicious Score calculated by the classifier trained in each additional round, respectively. The “True Label” shown at the top of the figure is the true label provided to its domain name, which is +1 for malicious and -1 for benign. Therefore, when the sign of the Malicious Score matches the sign of the true label, it indicates that the domain name can be correctly classified by the classifier. On the other hand, if the sign of the Malicious Score continues to change or does not correspond to the true label throughout the round, it means that the classifier cannot correctly classify the target domain name. Fig. 2.11(a) shows an example in which a malicious domain name can be correctly classified by active learning, Fig. 2.11(b) shows an example in which a benign domain name can be correctly classified by active learning, and Fig. 2.11(c) shows an example of a domain name which cannot be correctly classified regardless of which method we use to add training data. Considering the results in Fig. 2.11(a) and Fig. 2.11(b), the sign of the Malicious Score calculated by the classifier trained by random sampling either does not correspond to the sign of the true label or continues to change throughout the round. On the other hand, the sign of the Malicious Score calculated by the classifier trained by active learning differs from the sign of the true label at the beginning of the round, but as the round progresses, it corresponds to the sign of the true label. This indicates that domain names that cannot be correctly classified regardless of the amount of training data added during random sampling can be correctly classified by adding a small amount of training data by active learning. However, in the case of active learning, the Malicious Score fluctuates considerably with the addition of training data, and the classification result may differ from that of the previous round. However, these fluctuations in the Malicious Score can be overcome by ensemble learning, and the classification accuracy is stable.

In addition, as shown in Fig. 2.11(c), the existence of a domain name that cannot be classified even by adding training data by any method was confirmed. A characteristic of these domain names is that, despite being benign, it was posted on public blacklists such as Hphosts in the past, or even though it is a malicious domain name, it appears on Alexa Top Sites. In such a case, the feature of the domain name and the label provided to the domain name are inconsistent. This seems to be the reason for the continuous miss-classification.

2.5 Discussion

First, we discuss the two limitations of the proposed method. As indicated in 2.4.2, the classifier was found unable to correctly classify a particular domain name, regardless of the number of training data used for training. Therefore, for example, when an attacker creates a malicious web site and intentionally earns the number of web accesses to make it seem to be a popular site, the classifier may classify the domain name of the malicious web site as being benign. Domain names such as these are difficult to correctly classify by supervised machine learning. It may therefore be effective to create a whitelist and blacklist for each of these domain names and take countermeasures.

The second limitation is that the active learning method proposed in this paper is based on the premise that domain names are classified into two classes, and cannot be used for multi-class classification. Therefore, multi-class classification, i.e., to classify more than two types of labels other than malicious and benign, is impossible at present. In multi-class classification, margin sampling, least confident, and entropy-based sampling of uncertainty sampling adopted in this work are non-equivalent. Therefore, when attempting to align active learning, as proposed in this paper, to multi-class classification, it would be necessary to verify in advance which method is most effective and whether uncertainty sampling is effective.

Next, we consider the practicality of the proposed method. In this paper, all domain names in our data sets are labeled in advance for the convenience of evaluating the proposed method, i.e., the active learning in our experiments regards the operation of restoring hidden labels as a pseudo-human labeling operation. In other words, we do not carefully examine the time and labor required to actually label a domain name. However, by applying the proposed method involving both active learning and ensemble learning, it is shown that the number of domain names to be labeled by humans can be reduced to approximately one hundredth. Therefore, a simple calculation shows that the time required for human labeling is reduced by one hundred times, and the burden due to labeling is obviously reduced.

In addition, when using the proposed ensemble learning method, it is necessary to train a total of T classifiers and to predict the labels of the domain names to be classified in all the classifiers trained. Therefore, compared with the case where the detection system is operated with one classifier, the time required for training the classifier and the time required for the classifier to classify the domain name to be classified are T times. Therefore, we measured the time taken for such classifier training and the time taken for domain name classification by the classifier. The experiments were conducted on a Linux server with Intel Xeon E 5-2660 v3 CPU and 128 GB memory. The measurement results indicated that the time required for training one classifier was 0.04 seconds on average, which was a very short time. Furthermore, the average time required for one classifier to classify 20,000 domain names, i.e., the test data, was 0.018 seconds. Therefore, the time required for classification per domain name is only $0.018[s] \div 20,000 = 0.9[\mu s]$. Thus, we confirmed that the classifier training time and the time it takes for the classifier to classify a domain name do not constitute a bottleneck.

2.6 Related Work

2.6.1 Malicious Domain Name Detection System Using Supervised Machine Learning

Many systems based on supervised machine learning have been proposed to detect malicious domain names used in cyber-attacks. In this section, we describe representative systems. Notos [4] is the earliest malicious domain name detection system that was proposed. Notos extracts a feature from the network characteristics of IP addresses used by past malicious domain names and the character string characteristics of the domain names themselves, and classifies malicious and benign domain names using a decision tree. Exposure [5] is a system that extracts the time change of DNS query traffic from a large-scale user as a feature and uses this for the early detection of a malicious domain name using a decision tree. Kopis [6] is a system that uses a Random Forest to classify a malicious domain name by using the query behavior from the user to the domain name observed on the authoritative DNS server of the top level domain (TLD) as a feature. Pleiades [2] uses the characteristics of abnormal DNS queries for domain names that do not exist on the caching DNS server as a feature and uses decision trees and Hidden Markov Model (HMM) to classify the malicious domain name generated by DGA. Segugio [7] is a system that focuses on the DNS traffic patterns of malware infected users in a large-scale network typified by ISP and specifies malicious domain names similar to known malicious domain names. Predator [3] monitors the registration information of the domain name on the registry of a certain TLD, extracts the registered information characteristic similar to the past malicious domain name as a feature, and classifies new malicious domain names by using the Convex Polytope Machine (CPM), which is an ensemble of linear discriminators. DomainProfiler [8] applies a Random Forest using both a feature based on time series changes of usage forms of past malicious and benign domain names and a feature obtained from IP addresses used by each domain name to classify the malicious domain name early.

In our work, we showed that the classification accuracy is improved by applying active learning and ensemble learning to DomainProfiler. However, these two learning approaches can be applied to methods other than DomainProfiler as well, and it would be expected to also be effective for methods using supervised machine learning requiring labeled data as shown in this section.

2.6.2 Application of Active Learning to Security Problems

A few examples in which active learning was applied to security problems were also reported. Moskovitch et al. [20] used active learning as a method for efficiently adding unknown malicious code to training data to train a classifier that identifies malicious code. Zhao et al. [21] proposed cost-sensitive online active learning (CSOAL), which takes into consideration both malicious-benign-imbalance problems and constraints on the number of training data that can be labeled in malicious URL detection. However, the disadvantage of CSOAL is that the classifier is required to be a linear regression model. Beaugnon et al. [22] proposed ILAB, a strategy for constructing a malware intrusion detection system using active learning. By using class information in addition to benign and malicious label information, ILAB not only labels data near the classification decision boundary but also data included in the rare category. Specifically, when classifying with the Latest Classifier, data that are near the decision boundary of classification, that are ambiguous in terms of class information located near the boundary line between classes, and that are clear in terms of class information located near the center point of the class, are selected and labeled.

As mentioned above, although active learning has previously been applied to security problems, as far as the authors know, active learning has not yet been applied to malicious domain name detection. This paper is the first to propose active learning as a method for selecting a domain name to be labeled and to propose a practical method for constructing and operating a malicious domain name detection system. Furthermore, this work is the first to use ensemble learning in combination to avoid deterioration of the generalization ability of the classifier concerned with active learning.

2.7 Conclusion

When operating a system that detects malicious domain names using supervised machine learning, it is necessary to label new domain names of which the status (malicious or benign) is unknown and to acquire new training data. In this paper, we proposed a method that uses active learning to label domain names close to the classification decision boundary between benign and malicious. In addition, we proposed a method based on ensemble learning that integrates each new classifier that is trained using sequentially added training data with classifiers that have previously been trained using earlier training data and then outputs the final prediction results. We applied these proposed methods to actual domain name data based on a feature defined by the malicious domain name detection system DomainProfiler and evaluated its effectiveness.

Our experiments to evaluate the proposed method (compared to the existing method, which labels randomly selected data) showed that the classification accuracy can be improved by using a very small amount of training data and that the fluctuation in the classification accuracy is reduced and stabilized. As a result, we acquired a classifier with high classification accuracy and generalization ability, while reducing the high cost of labeling.

We also analyzed training data added by active learning and confirmed that domain names with various feature were added in equal amount as training data. We consider that this is the reason why active learning is effective.

Furthermore, apart from evaluating the classification accuracy of the entire set of test data, the variation of the classification result of each domain name in the test data was also investigated. This enabled us to clarify the domain names the proposed method is able to effectively classify and those that are difficult to classify by machine learning.

3

Large-scale Analysis of Cloud Service Abuse

3.1 Introduction

A cloud service is an infrastructure that is designed to provide users with a required amount of computing resources such as servers, storage, and applications, which are owned by the cloud service provider and lent as needed. Due to the considerable convenience, the global market for cloud services continues to expand rapidly. It has been shown that the size of the global market for cloud services increased by 37.6% to $26.3 billion in the second quarter of 2019 from $19.1 billion in the previous year. However, cloud services can also be abused as an infrastructure for cyber-attacks. For example, a large number of cloud servers were used for a brute-force attack to hijack Instagram accounts, where the servers on the cloud were used as a command and control (C&C) server [23, 24]. In this paper, we refer to cyber-attacks that abuse cloud services as "cloud service abuse."

Cloud service abuse poses risks for both legitimate users and cloud service providers. For example, in the cloud-based email-sending service Amazon simple email service (SES), a legitimate user could not send emails because the IP address assigned to the user was blacklisted [25]. This is because the IP address was shared between multiple users in the cloud service and some of the users might be involved in cyber-attacks. Thus, there is a risk that users will be subjected to various restrictions if IP addresses that have been blacklisted for malicious activity in the past are assigned to the servers that they use. As a result, there is also the risk that the reputation of cloud service providers will decrease. However, the actual status of cloud service abuse has not yet been clarified.

Therefore, to the best of our knowledge, this study represents the first large-scale analysis revealing the actual situation of cloud service abuse and discussion of effective countermeasures. To conduct this analysis, we required a method for large-scale observation of cloud service abuse that involves various types of cyber-attacks. To perform this, we focused on using different types of blacklists in combination. Then, we conducted a large-scale analysis of cloud service abuse for four typical cloud services: Amazon Web Service Elastic Compute Cloud (AWS), Microsoft Azure (Azure), Google Cloud Platform (GCP), and Oracle Cloud (Oracle). In our analysis, using 45 blacklists for 81 days, a total of 32,743 blacklisted IP addresses from cloud service providers were observed.

We also discovered five different aspects of cloud service abuse: (1) changes in the number of blacklisted cloud IP addresses over time, (2) the types of attacks involved in cloud service abuse, (3) trends regarding IP address regions, (4) differences in on-list duration of the blacklisted IP addresses depending on the attack type and cloud service provider, and (5) the status of deregistration of blacklisted cloud IP addresses. These findings suggest that cloud service providers need to detect abuse of their services early and take appropriate countermeasures.

The contributions of this study are as follows.

(1) We conducted the first large-scale analysis of cloud service abuse.

(2) We proposed an observational method for cloud service abuse using many diverse blacklists.

(3) We revealed the actual status of cloud service abuse.

(4) We discussed countermeasures against cloud service abuse for cloud service users, cloud service providers, and blacklist providers.

3.2 Cloud Services and Blacklists

Figure 3.1 shows the relationship between cloud service users, cloud service providers, and blacklist providers in a situation where cloud service abuse occurs. As shown in Fig. 3.1, when an attacker abuses the cloud service, the cloud IP addresses involved in the attack may be blacklisted, which causes false restrictions on legitimate users. In this section, we describe the cloud service (Section 3.2.1) and the blacklists (Section 3.2.2).

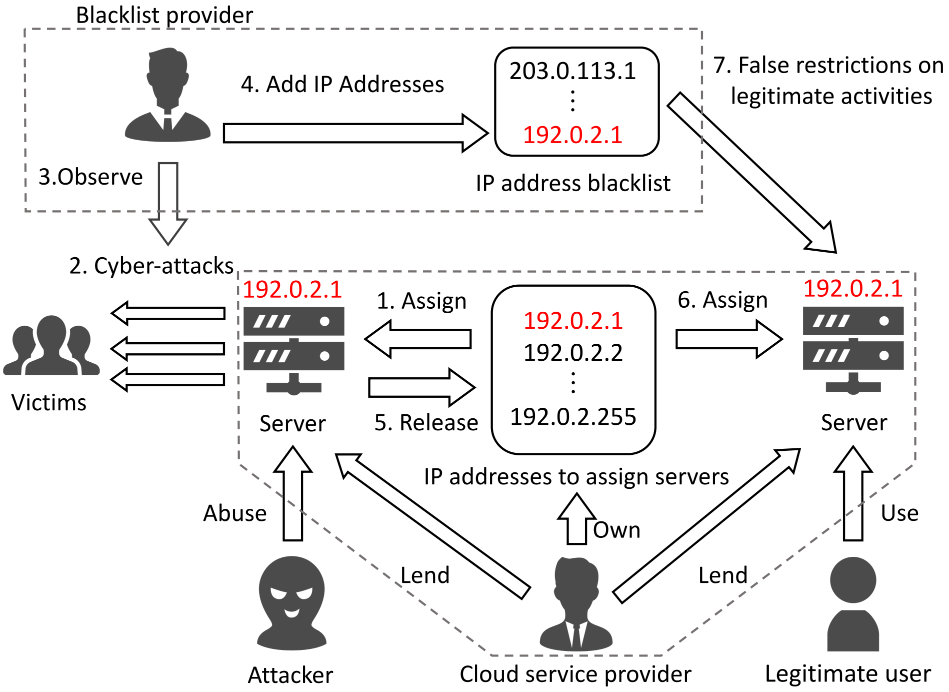


Fig 3.1: Stakeholders of cloud service abuse.

3.2.1 Cloud Services

With cloud services, users can reduce the initial investment, maintenance, and operational costs associated with physical hardware. In general, cloud service providers have data centers in multiple regions around the world. When using cloud services, the user can select the region where they want the computing resources to be located. The prices and range of IP addresses assigned to the server vary depending on the selected region. IP addresses in cloud services are shared between users. Thus, the IP address assigned to the server is released when the server is shut down and it is assigned to another server.

There are many cloud service providers. In this study, we focused on AWS, Azure, GCP, and Oracle for our measurements. There are two reasons for selecting these four cloud services. First, these are typical/popular cloud services and are considered to be abused by more attackers. Second, in these cloud services, the range of public IP addresses assigned to the server is available to the public [26, 27, 28, 29]. This allowed us to determine whether a given IP address was used by a cloud service.

3.2.2 Blacklists

A blacklist is a list of IP addresses that have been found to be involved in malicious activity. Blacklists are used to identify communications that use the blacklisted IP addresses as the source or destination. Blacklists are not static but are updated regularly by the blacklist provider. However, the update time and interval differ for each blacklist. In addition, the period for which a blacklisted IP address remains on the blacklist differs for each blacklist. This is because the listing policies of blacklist providers are different.

Some blacklists allow third parties to apply for the deregistration of blacklisted IP addresses. There are two main reasons why this is possible. One is that the IP addresses of users who did not originally perform malicious activities may be falsely blacklisted. Another reason is that the blacklisted IP address may have already stopped performing malicious activities due to subsequent countermeasures.

3.3 Measurement Method

We conducted a large-scale analysis of cloud service abuse for four typical cloud services: AWS, Azure, GCP, and Oracle. For this analysis, it was necessary to have an environment where large-scale, continuous, and direct observation of cloud service abuse involving various types of attacks was possible. However, it is extremely difficult to prepare such an environment for observing attacks directly. For example, honeypots and darknets could be potential observation methods for cloud service abuse. However, these are restricted by limited observation ranges, and the types of attacks are constrained to brute-force attacks and scans.

In this study, we focused on blacklists of IP addresses to solve this problem. By integrating different types of blacklists, both the observation range and types of observable attacks were increased. In addition, as blacklists are updated regularly, it is easy to conduct a time series analysis. Conventional studies on blacklists [30, 31, 32] have focused on the accuracy and characteristics of blacklists themselves. We used multiple blacklists as the practical method for extensively observing cloud service abuse without direct observation. This is a significant difference versus conventional studies on blacklists and it also represents the technical importance of our measurement method.

3.3.1 Observation of Cloud Service Abuse

This section describes the observation method for cloud service abuse using blacklists. The method consists of three steps: acquiring blacklists, classifying blacklists according to attack type, and extracting the IP addresses of cloud service providers.

Acquiring Blacklists. A total of 45 public blacklists were acquired from 22 different blacklist providers once per day at the same time. As a result of investigating the update frequency of

Table 3.1: Provider name, number of blacklists from each provider, and total number of unique blacklisted IP addresses.

#	Provider Name	# Lists	# IP Addresses
1	Badips	11	646,370
2	Fail2ban	7	316,499
3	Normshield	6	174,514
4	ProjectHoneyPot	3	9,277
5	AlienVault	1	228,799
6	Bambenek	1	2,817
7	BinaryDefense	1	19,639
8	BotScout	1	52,158
9	CleanTalk	1	451,070
10	CyberCrime	1	1,645
11	Dangerrulez	1	4,082
12	DShield	1	43,706
13	Feodo	1	1,235
14	Haley	1	59,041
15	LashBack	1	989,571
16	MyIP	1	5,119
17	NixSpam	1	250,166
18	Nothink	1	31,806
19	Sblam	1	35,083
20	StopForumSpam	1	178,415
21	Talos	1	2,620
22	VoIPBL	1	90,009

each blacklist, we decided to acquire blacklists once a day. The acquisition period lasted 81 days from June 30, 2019 to September 18, 2019. The blacklists acquired in this study were also used in [32, 33], which compared and analyzed the characteristics of multiple IP address blacklists. We selected blacklists that were continuously updated during the acquisition period.

For the acquired blacklists, Table 3.1 summarizes the names of the blacklist providers, the number of blacklists from each provider, and the total number of unique blacklisted IP addresses. Note that four blacklist providers created multiple blacklists, where the remaining providers each created one blacklist.

Table 3.2: Attack type, number of corresponding blacklists, and total number of unique blacklisted IP addresses.

Attack Type	**# Lists**	**# IP Addrs**
Scan	5	110,465
Brute-force	10	514,735
Malware	3	5,697
Exploit	10	249,628
Botnet	7	151,204
Spam	10	2,135,486
Total	45	3,167,215

Classifying Blacklists by Attack Type. Next, we describe the procedure for classifying the acquired blacklists according to attack type. The 45 acquired blacklists were classified into 6 attack types based on explanations from the blacklist providers: Scan, Brute-force, Malware, Exploit, Botnet, and Spam. These six attack types are defined in [32]. This categorization enabled us to conduct an analysis that considers what type of attack each blacklisted IP address performed. Here, hosts performing port or vulnerability scans are classified as Scan, hosts making brute-force login attempts are classified as Brute-force, malware C&C and distribution servers are classified as Malware, hosts attempting to remotely exploit vulnerabilities are classified as Exploit, compromised hosts belonging to a botnet are classified as Botnet, and hosts sending spam are classified as Spam. For example, a blacklist with the explanation “IP address confirmed to have a brute-force attack in a honeypot” is classified as Brute-force, whereas a blacklist with the explanation “IP address confirmed to be the sender of spam” is classified as Spam. Table 3.2 lists the number of blacklists classified into each attack type and the total number of unique blacklisted IP addresses from all classified blacklists for each attack type.

Extracting the IP Addresses of Cloud Service Providers. Finally, we describe the procedure for extracting the IP addresses of cloud service providers from the blacklisted IP addresses. As explained in Section 3.2.1, each of the four cloud services selected in this study releases a range of public IP addresses that can be assigned to servers lent to users [26, 27, 28, 29]. Note that we obtained GCP public IP addresses range from the DNS record associated with *_cloud-netblocks[1,2,3,4,5].googleusercontent.com* as of September 18, 2019. We compared and matched the blacklisted IP addresses and the above range of public IP addresses. As a result, the blacklisted IP addresses of the cloud service providers were extracted. By observing the blacklisted cloud IP addresses, it was possible to indirectly observe the occurrence of cloud service abuse. In this study, we investigated only IPv4 addresses. This is because the blacklists acquired in this study contained a very small number of IPv6 addresses. The total number of unique blacklisted IPv6 Addresses was 714 (approximately 0.023% of the total).

3.3.2 Analysis of Trends in Cloud Service Abuse

In this section, we describe the technique for analyzing trends in cloud service abuse using blacklists. We integrated the 45 blacklists acquired on the same day into one list. The IP addresses in the integrated list were blacklisted in at least one of 45 blacklists that was used for integration. As the blacklists were acquired for 81 days, 81 integrated lists were created. Using these 81 integrated lists, we investigated when and which cloud IP address was blacklisted and the attack type of the blacklists in which it was registered. This integrated list is useful for identifying the daily trends of cloud service abuse.

We analyzed cloud service abuse from the following six viewpoints.

- How many IP addresses are abused per day and how do they change over time? (Section 3.4.1)

- Is there a difference in the number of abused IP addresses depending on the cloud service provider and type of attack? (Section 3.4.1 and 3.4.2)

- Are there any regional characteristics of the abused IP addresses? (Section 3.4.3)

- How long do abused IP addresses appear on the blacklist? (Section 3.4.4)

- Can we observe applications from cloud service providers or users for the deletion of IP addresses from blacklists? (Section 3.4.5)

- Did the IP addresses observed in the blacklists actually perform cyber-attacks? (Section 3.4.6)

3.4 Measurement Results

3.4.1 Number and Changes of Blacklisted IP Addresses

For each cloud service provider, we investigated the number of IP addresses registered in at least one of the 45 acquired blacklists. Approximately 7,137 AWS, 2,733 Azure, 1,660 GCP, and 337 Oracle IP addresses were blacklisted daily. Moreover, approximately 397 AWS,

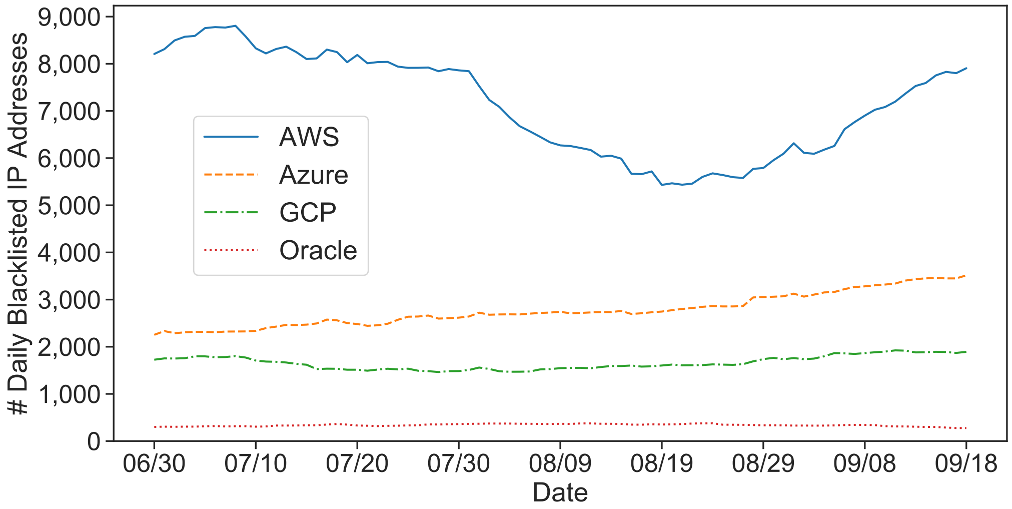


Fig 3.2: Change in the number of daily blacklisted IP addresses.

100 Azure, 66 GCP, and 13 Oracle IP addresses were replaced per day. In other words, approximately 12,000 cloud IP addresses were blacklisted and about 5% were replaced per day. Figure 3.2 shows a graph of date versus the number of cloud IP addresses registered in the blacklists acquired on that date. The change in the number of blacklisted IP addresses over time differs for each cloud service provider. The number of AWS IP addresses decreased from the observation start date of June 30 to the end of August, then increased after August. Additionally, throughout the observation period, the number of blacklisted Azure IP addresses increased slowly, and the number of blacklisted GCP and Oracle IP addresses remained almost unchanged. The total number of unique blacklisted IP addresses over the observation period was 22,122 for AWS, 5,753 for Azure, 4,121 for GCP, and 747 for Oracle.

3.4.2 Type of Attack

Next, we conducted an analysis of blacklisted IP addresses based on the type of attack. In Section 3.4.1, all 45 blacklists were integrated without considering the attack type. However, for the results discussed in this section, the blacklists were integrated according to attack type. We investigated the number of unique blacklisted IP addresses for each attack type and the proportion of these numbers with respect to the total number for all attack types. The results are listed in Table 3.3. Many IP addresses from each cloud service provider were registered as Brute-force and Spam, which accounted for 75% to 87% of the total blacklisted IP addresses. This suggests that brute-force attacks and spam-sending are often involved in cloud service abuse. On the other hand, each cloud service provider differed in terms of the proportions

Table 3.3: Number of blacklisted IP addresses for each attack type.

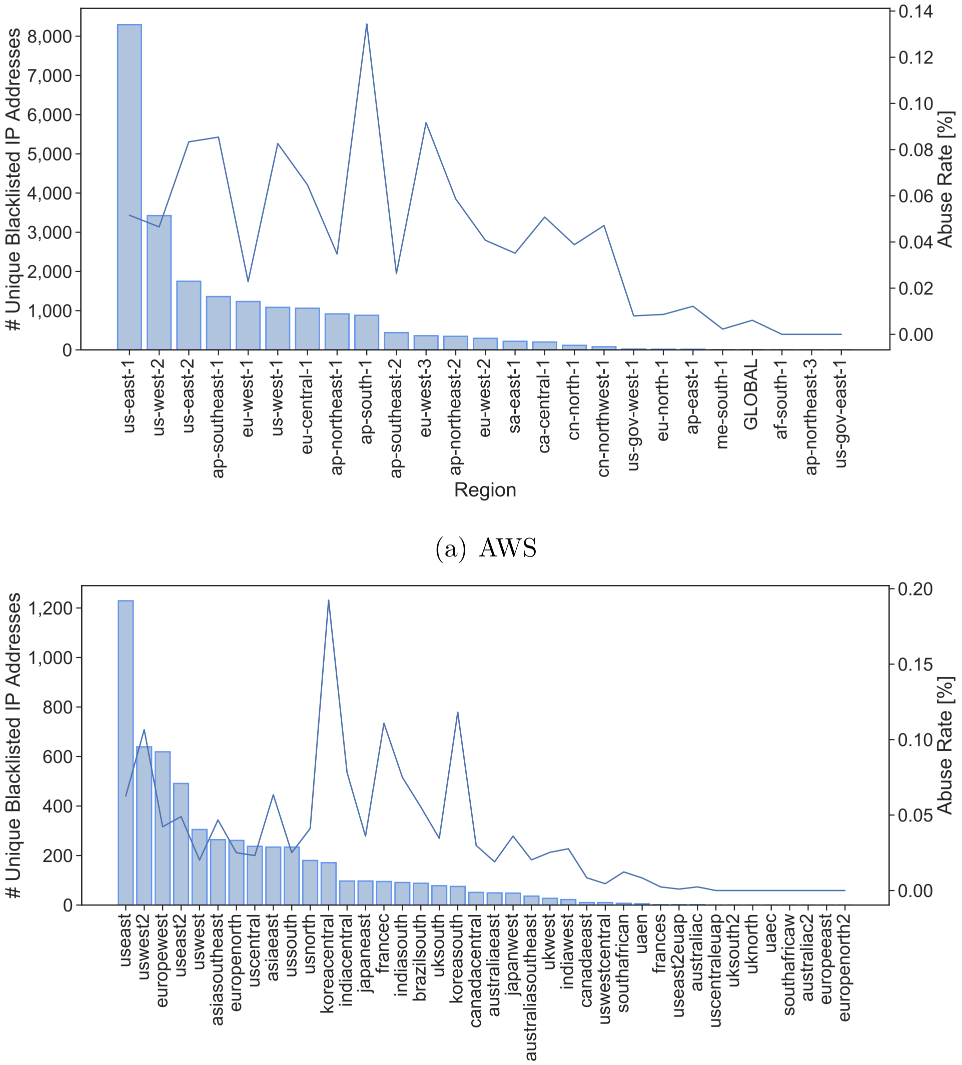
Attack Type	**#**	**AWS**	**Azure**	**GCP**	**Oracle**
Scan	5	386 (1.7%)	78 (1.3%)	89 (2.0%)	13 (1.7%)
Brute-force	10	6,723 (30%)	2,686 (45%)	1,451 (33%)	256 (33%)
Malware	3	174 (0.77%)	60 (1.0%)	39 (0.88%)	1 (0.13%)
Exploit	10	1,939 (8.6%)	971 (16%)	368 (8.3%)	77 (9.9%)
Botnet	7	1.312 (5.8%)	352 (5.9%)	606 (14%)	11 (1.4%)
Spam	10	12,063 (53%)	1,817 (30%)	1,901 (43%)	420 (54%)
Total	45	22,597 (100%)	5,964 (100%)	4,454 (100%)	778 (100%)

of each attack type. For example, the proportions of IP addresses used for Brute-force and Exploit in Azure and Botnet in GCP are larger than in the other cloud services.

The total number of blacklisted IP addresses in Table 3.3 does not coincide with the total number of unique blacklisted IP addresses shown in Section 3.4.1. This is because some IP addresses were registered in blacklists corresponding to 2 or more attack types. These IP addresses are considered to be associated with multiple types of cyber-attacks. Many such IP addresses were registered in Spam blacklists and other blacklists such as Exploit and Brute-force. This suggests that cloud service abuse involving multiple types of cyber-attacks tends to exploit vulnerabilities remotely or perform brute-force attacks in addition to spam-sending.

3.4.3 IP Address Regions

In this section, we investigate the regions of blacklisted IP addresses from cloud service providers. We selected AWS and Azure as cloud service providers to analyze regions because these two providers publish information linking IP addresses and regions and have a sufficient number of both. We investigated 22,122 AWS and 5,753 Azure IP addresses, the total number of unique blacklisted IP addresses over the observation period clarified in Section 3.4.1. In addition to the number of blacklisted IP addresses for each region, we investigated the proportion of these numbers to the total number of IP addresses in each region. In this paper, this proportion is called the IP address “abuse rate” for each region. The total number of IP addresses in each region was derived based on the following procedure. First, we expanded the IP address range (CIDR notation) in each region by IP address, excluding the network address and broadcast address. However, in the case of /31 as an exception, the network address and broadcast address were also included. Then, we calculated the total number of deployed IP addresses.



(a) AWS

(b) Azure

Fig 3.3: Number of unique blacklisted IP addresses and abuse rate for each region.

Figure 3.3 shows the graphs of region versus number of unique blacklisted IP addresses and abuse rate for AWS and Azure. Both AWS and Azure have a bias in the number of unique blacklisted IP addresses among regions. In AWS, US regions, except for *us-west1*, have low prices and fast communication speeds [34]. Therefore, it is highly likely that attackers selected inexpensive regions, like general users. Similarly, in Azure, there are many blacklisted IP addresses in US regions with low prices. However, the number of blacklisted IP addresses associated with *europewest*, which has a relatively high price, is also large. That is, the regions are not selected based only on price. Here, we referred to [35] for the price of each region in Azure.

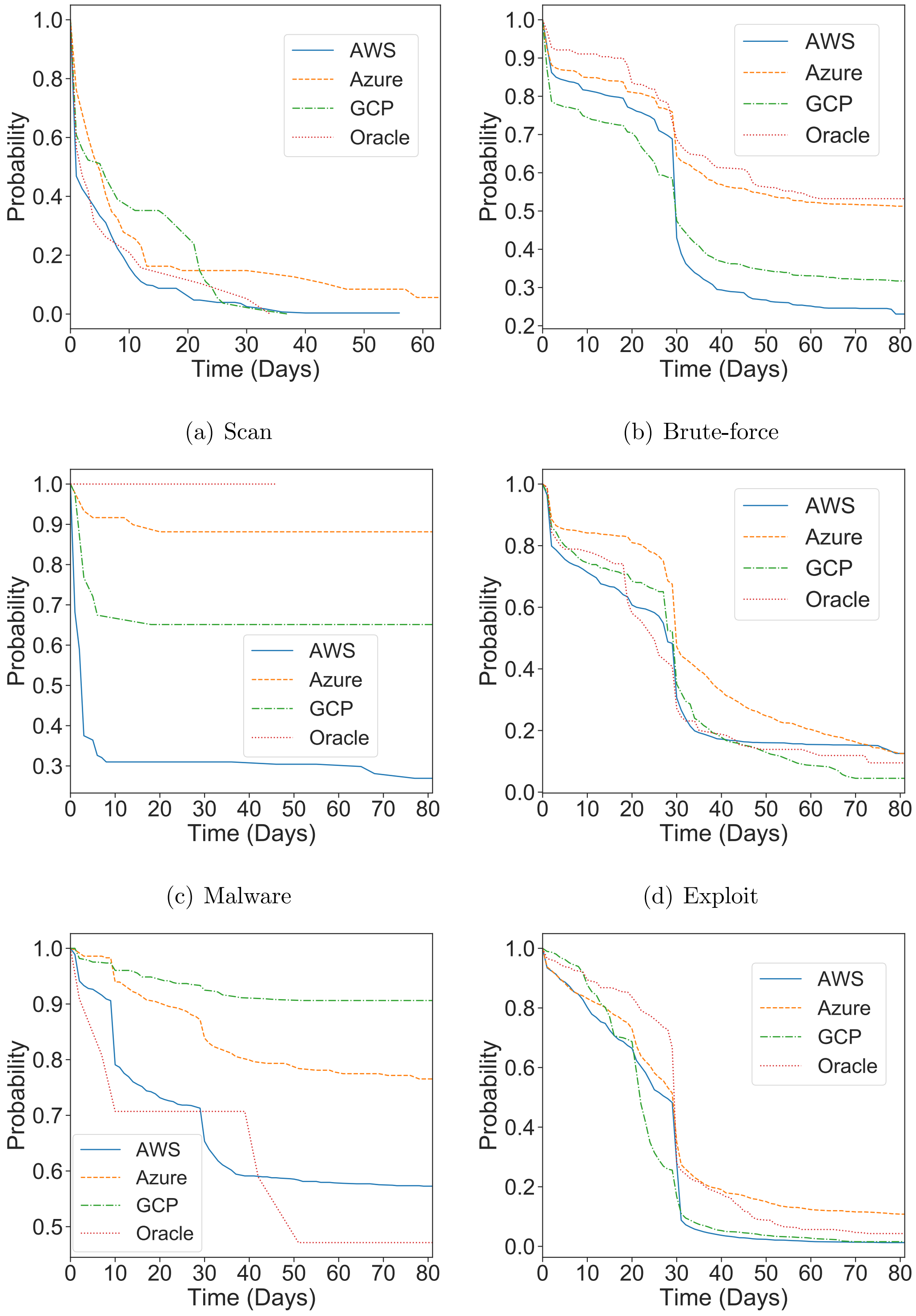
Similar to the number of blacklisted IP addresses, there is also a bias in the abuse rate for each region. In addition, the abuse rates of regions with many blacklisted IP addresses are not necessarily high. To quantitatively evaluate this observation, we use Spearman's rank correlation coefficient, which calculates the correlation between the rank of the number of blacklisted IP addresses and the rank of the abuse rate. The calculated Spearman's rank correlation coefficient is approximately 0.41 for AWS and 0.14 for Azure. This shows that the number of blacklisted IP addresses and the abuse rate have a slightly positive correlation for AWS and little correlation for Azure. Examples of regions where the rank of the abuse rate is higher than that of the number of blacklisted IP addresses are *ap-south-1* (India) and *eu-west-3* (France) for AWS and *koreacentral* and *koreasouth* (Korea), *francec* (France), and *indiacentral* and *indiasouth* (India) for Azure. In particular, the India and France regions have high abuse rates for both AWS and Azure, which suggests that the proportion of IP addresses that have been abused by cyber-attackers tends to be large in these regions.

3.4.4 Probability that IP Address Continues to be Blacklisted

In this section, we define the probability that an IP address continues to be on a blacklist for N days after first being listed as "on-list duration probability for N days." We analyzed the on-list duration probability using the Kaplan–Meier method [36]. In this study, the observation period was limited to 81 days. Therefore, some IP addresses continue to be blacklisted even on the observation end date. Using the Kaplan–Meier method, we can calculate the on-list duration probability even under this condition. We calculated the on-list duration probability for each type of cyber-attack and for each cloud service provider.

The six graphs in Fig. 3.4 show the results of the Kaplan–Meier analysis. For each graph, the vertical axis indicates the on-list duration probability (complementary cumulative distribution) and the horizontal axis indicates the elapsed days since the IP address was first blacklisted. If the on-list duration probability remains high as time passes, the cloud service abuse continued for a long time, which means that the cloud service provider is not adequately addressing the abuse.

From Fig. 3.4, we can confirm that the on-list duration probability is different for each type of cyber-attack and each cloud service provider. For example, the on-list duration probability of (a) Scan, decreases steadily, where those of (c) Malware and (e) Botnet decrease slowly and converge to a high lower limit, and those of (b) Brute-force, (d) Exploit, and (f) Spam decrease significantly after 30 days. Especially, for (b) Brute-force, AWS and GCP have lower on-list duration probabilities than Azure and Oracle, and for (c) Malware, only AWS has a lower on-list duration probability compared with the other cloud service providers.



(a) Scan

(b) Brute-force

(c) Malware

(d) Exploit

(e) Botnet

(f) Spam

Fig 3.4: On-list duration probability of blacklisted IP address for each attack type and cloud service provider.

Here, we analyzed why the on-list duration probability significantly decreases for (b) Brute-force, (d) Exploit, and (f) Spam for elapsed times greater than 30 days. Many of the blacklists classified into these attack types specify a policy wherein IP addresses remain on the blacklist for at least 30 days after the final observation. In other words, cyber-attacks were not observed after the first blacklisting, and many IP addresses were removed from the blacklists immediately, once the blacklisting period exceeded 30 days according to this policy. If the attack is observed again N ($N \leq 30$) days after the first blacklisting, the blacklisting period will continue for at least $30 + N$ days.

3.4.5 Deregistration of Blacklisted Cloud IP Addresses

We also investigated the deregistration of blacklisted cloud IP addresses. The legitimate user or cloud service provider can apply the deregistration of the blacklisted cloud IP address that are no longer involved in cloud service abuse. By investigating the status of deregistration of blacklisted cloud IP addresses, we can estimate the current status of countermeasures against cloud service abuse. If the blacklisted IP address is removed before the policy-defined period (e.g., 30 days), there was considered to be an application for deregistration from a third party. As a result of the investigation, we identified the cloud IP addresses that were considered to be removed according to an application for deregistration. Specifically, in a blacklist with a policy of on-list duration of 30 days, 361 IP addresses were deleted within 30 days. For ethical considerations, we do not mention this blacklist provider's name. This is less than 5% of the total number of unique cloud IP addresses in the blacklist. This means that most of the blacklisted IP addresses were continuously used for cyber-attacks or that there was no application for the deregistration of blacklisted IP addresses that were no longer being used for cyber-attacks.

3.4.6 Observation of Cloud Service Abuse Using Darknet

To evaluate the reliability of the results of our analysis, it was necessary to verify whether the blacklisted cloud IP addresses were actually involved in cyber-attacks. To this end, we needed to observe cloud service abuse using methods other than blacklists. In this study, we used the darknet. The darknet is a space of reachable and unused IP addresses on the Internet, and almost all packets arriving at the darknet can be regarded as malicious [37]. In this section, we discuss the source IP addresses of packets arriving at the darknet and investigate how many

Table 3.4: Number and coverage of blacklisted IP addresses observed in darknet.

	AWS	**Azure**	**GCP**	**Oracle**
# Blacklisted IP Addresses	269	71	83	18
# Observed in Darknet	202	60	80	18
Coverage	75%	85%	96%	100%

are cloud IP addresses. We focus on Scan, which is the only attack type that the darknet can observe.

The darknet used in this study was the UCSD network telescope [37]. This darknet is an /8 network that has over 16 million IP addresses. We investigated the packets arriving at this darknet for 32 days from August 7 to September 7, 2019. The number of unique packet source IP addresses observed during this period was 109,127,891. These IP addresses included 120,170 AWS, 13,156 Azure, 8,014 GCP, and 621 Oracle IP addresses.

Next, we compared these IP addresses and the unique IP addresses registered in blacklists classified as Scan during the same period. By this matching, we verified whether the blacklisted IP addresses actually conducted the scan. The matching results are shown in Table 3.4, where coverage means the proportion of blacklisted IP addresses that were observed in the darknet. The results show that the lowest coverage was 75% for AWS. In other words, many of the blacklisted cloud IP addresses were observed in the darknet in the same period. This shows that many cloud IP addresses that were registered in the blacklists of Scan were involved in cyber-attacks.

3.5 Discussion and Limitations

3.5.1 Discussion

From the discussion in Section 3.4, we confirmed that the seriousness of cloud service abuse cannot be ignored and that effective countermeasures are necessary. In this section, we make the following suggestions for cloud service users, cloud service providers, and blacklist providers.

Cloud Service Users. When using a cloud service, users should check whether the assigned IP address is blacklisted. If the assigned IP address was previously used for cloud service abuse, the IP address could still be blacklisted, and the communication could be blocked. For example, using the web service IPVoid [38], we can compare against a total of over 100 blacklists just by entering the IP address we want to check in the browser. If the IP address is blacklisted, users can take measures such as receiving a different IP address by stopping and restarting the server.

Cloud Service Providers. In situations where cloud service abuse occurs constantly, if an abused IP address is released and immediately assigned to another user, there are various restrictions on that user's service. Therefore, cloud service providers should detect and manage cloud service abuse at an early stage to minimize cyber-attacks using their services to provide users with highly available cloud services. To achieve this, cloud service providers can use the measurement method conducted in this study. That is, collect multiple IP address blacklists, find their blacklisted IP addresses, and take action to warn or suspend the corresponding malicious user.

Blacklist Providers. Blacklist providers need to reduce false positives as much as possible when creating blacklists. In Section 3.4.4, we showed that many IP addresses continue to be blacklisted for 30 days according to policy but are considered to have conducted no attacks after the first blacklisting. Because the IP addresses of cloud service providers are shared among users, it is not desirable to keep such IP addresses blacklisted for a long time. Using the measurement method in this study, blacklist providers can identify cloud IP addresses and treat them differently than other IP addresses. For example, when an attack from a cloud IP address is observed, not only blacklisting but also informing the cloud service provider may lead to a faster response.

3.5.2 Limitations

There are two main limitations to this study. One is that the observation range of cloud service abuse is limited to the observation range of blacklist providers. As shown in Section 3.4.6, there are a large number of IP addresses that were observed in the darknet but were not blacklisted. Because it is impossible to observe all possible attacks on the Internet, it is not easy to completely solve this limitation. However, in the future, we would like to conduct an analysis that is as comprehensive as possible by increasing the number of acquired blacklists and days of investigation. The other is that the accuracy of the measurement method used in this study depends on the accuracy of the information provided by the blacklist provider. However, as shown in Section 3.4.6, we confirmed that highly accurate information was provided at least by the blacklists classified as Scan. Additionally, we will strive to improve the accuracy by combining as many types of blacklists as possible.

3.6 Related Work

IP addresses are the most basic and essential identifier on the Internet, and therefore many related studies have been conducted. In this section, we summarize related works that are roughly divided into studies that focused on malicious IP addresses used for attacks and studies that focused on changes in IP addresses themselves.

Malicious IP Addresses. Ramachandran et al. [39] analyzed the characteristics of the source IP addresses of a large amount of spam mail collected by spam traps from 2004 to 2005 and showed that the source addresses of spam mail were biased to a specific IP address range. Moreover, they showed that commercial IP blacklists for spam mail countermeasures cannot identify more than 30% of spam mail source IP addresses, and such source IP addresses are not blacklisted for more than one month [30]. Metcalf et al. [31] showed that there are many unique malicious IP addresses for each blacklist and that there is less duplication of IP addresses among blacklists by collecting and investigating multiple blacklists that registered malicious IP addresses. In 2019, Li et al. [32] collected a large number of public and commercial IP address blacklists and proposed objective evaluation indicators. Based on the indicators, they proved that the current IP blacklist was still insufficient for protecting users and organizations.

Our study is based on multiple malicious IP addresses or IP blacklists, which is similar to the above studies. However, there are two major differences. First, in our study, the investigated type of attack was not limited to spam emails but instead considers the more general current trend of cyber-attacks. Second, we focused on the IP addresses of cloud service providers and revealed the actual status of abuse specific to cloud services.

Changes in IP Addresses. Liu et al. [40] defined a DNS record that remains even though the DNS record reference resource (e.g., domain name or IP address) has not been used and is invalidated as a dangling DNS record (Dare). This was the first study to identify the security risks of Dare. They clarified that when a Dare reference destination is an IP address from a cloud service, a third party can obtain it after it is released. Pariwono et al. [41] verified the same problem as [40] by focusing on the domain name and IP addresses referenced from an Android application and revealed its risks. Nakamori et al. [42] emphasized that the same IP address is not always assigned to the same user when a dynamic IP address is assigned by an ISP or when an IP address is assigned by a cloud service provider. They proposed a method to identify such changeable IP address regions from the continuity of PTR records.

Considering the nature and actual status of changes to the owners of cloud IP addresses or dynamic IP addresses as shown in the above studies, we used 45 blacklists and analyzed the changes in malicious cloud IP addresses. We clarified the characteristics of malicious cloud IP addresses that continue to be blacklisted and the attack trends unique to cloud services for the first time.

3.7 Conclusion

In this paper, we conducted the first large-scale analysis of cloud service abuse. The main idea of our study was to use large and diverse blacklists for observing cloud service abuse without direct observation. Our analysis of four typical/popular cloud services using 45 blacklists over 81 days revealed the actual status of cloud service abuse: changes in the number of blacklisted cloud IP addresses over time, the types of attacks, trends regarding IP address regions, on-list duration of the blacklisted IP addresses, and the status of deregistration. The findings of this study provide a foothold for cloud service users, cloud service providers, and blacklist providers to effectively manage cloud service abuse.

4

Conclusion

When considering effective countermeasures against cyber-attacks, it is important to analyze communication data on the Internet. In this thesis, we focused on "address information" such as malicious domain names and malicious IP addresses used by attackers. By analyzing them in detail and extracting their features, we proposed and discussed effective countermeasures against evolving cyber-attacks. In Chapter 2, we introduced the concepts of active learning and ensemble learning to realize highly accurate detection of malicious domain names with only a small amount of training data. Active learning selects domain names that are considered useful for improving classification accuracy and labels only them, which leads to reducing labeling cost. Ensemble learning integrates the prediction results of multiple classifiers to improve classification accuracy and stability. As a result of the evaluation experiment, we revealed that our proposed method can achieve higher classification accuracy and stability with a much smaller number of training data than the conventional method. In Chapter 3, we conducted a large-scale analysis of cyber-attacks that abuse cloud services. We proposed a method for indirectly observing various cyber-attacks by combining many different types of IP Address blacklists. Our analysis of four typical/popular cloud services revealed the actual status of cloud service abuse: changes in the number of blacklisted cloud IP addresses over time, the types of attacks, trends regarding IP address regions, on-list duration of the blacklisted IP addresses, and the status of deregistration. As described above, the contributions of this thesis are to provide effective countermeasures and valuable insights for evolving cyber-attacks.

Acknowledgement

First, I am deeply grateful to Prof. Masato Uchida of Waseda University, School of Fundamental Science and Engineering, for instructing and supporting writing this thesis. Next, I truly feel grateful to Dr. Daiki Chiba and Dr. Mitsuaki Akiyama of NTT Secure Platform Laboratories, for their suggestive advice on the research and writing this thesis. Finally, I would also like to thank the members of Uchida laboratory who have assisted the research.

Bibliography

[1] Cisco. *Cisco 2016 Annual Security Report*. `http://mkto.cisco.com/rs/564-whv-323/images/cisco-asr-2016.pdf`. [Online; accessed 24-January-2020]. 2016

[2] Manos Antonakakis et al. "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware". In: *Proc. USENIX Security Symposium*. 2012, pp. 491–506.

[3] Shuang Hao et al. "PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration". In: *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2016, pp. 1568–1579.

[4] Manos Antonakakis et al. "Building a Dynamic Reputation System for DNS". In: *Proc. USENIX Security Symposium*. 2010, pp. 273–290.

[5] Leyla Bilge et al. "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis". In: *Proc. Network and Distributed System Security Symposium (NDSS)*. 2011.

[6] Manos Antonakakis et al. "Detecting Malware Domains at the Upper DNS Hierarchy". In: *Proc. USENIX Security Symposium*. 2011.

[7] Babak Rahbarinia, Roberto Perdisci, and Manos Antonakakis. "Segugio: Efficient Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks". In: *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2015, pp. 403–414.

[8] Daiki Chiba et al. "DomainProfiler: Discovering Domain Names Abused in Future". In: *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2016, pp. 491–502.

[9] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[10] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32.

[11] L. Breiman. "Bagging Predictors". In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140.

[12] R. E. Schapire. "The Strength of Weak Learnability". In: *Machine Learning* 5.2 (June 1990), pp. 197–227.

[13] *hpHosts*. `http://www.hosts-file.net/`. [Online; accessed 24-January-2020]

[14] *Alexa Top Sites*. `http://www.alexa.com/topsites`. [Online; accessed 24-January-2020]

[15] *VirusTotal*. `https://www.virustotal.com/`. [Online; accessed 24-January-2020]

[16] VERISIGN. *THE DOMAIN NAME INDUSTRY BRIEF*. `https://www.verisign.com/assets/domain-name-report-Q12019.pdf`. [Online; accessed 24-January-2020]. 2019

[17] Ponemon Institute. *The cost of malware containment*. `http://www.ponemon.org/library/the-cost-of-malware-containment`. [Online; accessed 24-January-2020]. 2015

[18] SECURITYWEEK. *How to Reduce False Positives and Move Faster on What Matters*. `https://www.securityweek.com/how-reduce-false-positives-and-move-faster-what-matters`. [Online; accessed 24-January-2020]. 2018

[19] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proc. ACM Knowledge Discovery and Data Mining (KDD)*. 1996, pp. 226–231.

[20] Robert Moskovitch, Nir Nissim, and Yuval Elovici. "Malicious Code Detection Using Active Learning". In: *Proc. Privacy, Security, and Trust in KDD (PinKDD)*. 2008, pp. 74–91.

[21] Peilin Zhao and Steven C. H. Hoi. "Cost-sensitive online active learning with application to malicious URL detection". In: *Proc. ACM Knowledge Discovery and Data Mining (KDD)*. 2013, pp. 919–927.

[22] Anaël Beaugnon, Pierre Chifflier, and Francis Bach. "ILAB: An Interactive Labelling Strategy for Intrusion Detection". In: *Proc. Research in Attacks, Intrusions, and Defenses (RAID)*. 2017, pp. 120–140.

[23] Hot for Security. *How any Instagram account could be hacked in less than 10 minutes*. https://hotforsecurity.bitdefender.com/blog/how-any-instagram-account-could-be-hacked-in-less-than-10-minutes-21409.html

[24] Cybers Guards. *Hackers abuse Microsoft Azure to use malware and evasion technology on C2 servers*. https://cybersguards.com/hackers-abuse-microsoft-azure-to-use-malware-and-evasion-technology-on-c2-servers

[25] Amazon Web Services, Inc. *Barracuda blocking email from SES*. `https://forums.aws.amazon.com/thread.jspa?messageID=897282`

[26] Amazon Web Services, Inc. *AWS IP Address Ranges.* `https://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html`

[27] Microsoft. *Microsoft Azure Datacenter IP Ranges.* `https://www.microsoft.com/en-hk/download/details.aspx?id=41653`

[28] Google Cloud. *Google Compute Engine FAQ.* `https://cloud.google.com/compute/docs/faq?hl=en`

[29] Oracle Cloud. *IP Address Ranges.* `https://docs.cloud.oracle.com/iaas/Content/General/Concepts/addressranges.htm`

[30] Anirudh Ramachandran, Nick Feamster, and Santosh Vempala. "Filtering spam with behavioral blacklisting". In: *Proc. ACM CCS.* 2007, pp. 342–351.

[31] Leigh Metcalf and Jonathan M. Spring. "Blacklist Ecosystem Analysis: Spanning Jan 2012 to Jun 2014". In: *Proc. ACM WISCS.* 2015, pp. 13–22.

[32] Vector Guo Li et al. "Reading the Tea leaves: A Comparative Analysis of Threat Intelligence". In: *Proc. USENIX Security.* 2019, pp. 851–867.

[33] FireHOL. *All Cybercrime IP Feeds.* `https://iplists.firehol.org/`

[34] Concurrency Labs. *Save yourself a lot of pain (and money) by choosing your AWS Region wisely.* `https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/`

[35] *Average Price Per Azure Region.* `https://azureprice.net/Region`

[36] Edward L Kaplan and Paul Meier. "Nonparametric estimation from incomplete observations". In: *Journal of the American statistical association* 53.282 (1958), pp. 457–481.

[37] CAIDA. *The UCSD Network Telescope.* `https://www.caida.org/projects/network_telescope/`

[38] IPVoid. *IPVoid.* `https://www.ipvoid.com/`

[39] Anirudh Ramachandran and Nick Feamster. "Understanding the network-level behavior of spammers". In: *Proc. ACM SIGCOMM.* 2006, pp. 291–302.

[40] Daiping Liu, Shuai Hao, and Haining Wang. "All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records". In: *Proc. ACM CCS.* 2016, pp. 1414–1425.

[41] Elkana Pariwono et al. "Don't throw me away: Threats Caused by the Abandoned Internet Resources Used by Android Apps". In: *Proc. ACM AsiaCCS.* 2018, pp. 147–158.

[42] Tomofumi Nakamori et al. "Detecting Dynamic IP Addresses and Cloud Blocks Using the Sequential Characteristics of PTR Records". In: *Journal of Information Processing* 27 (2019), pp. 525–535.