IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 September 2023, accepted 3 October 2023, date of publication 9 October 2023, date of current version 13 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323218

APPLIED RESEARCH

An Automatic Detection System for Fake Japanese Shopping Sites Using fastText and LightGBM

KEISUKE SAKAI[1], KOSUKE TAKESHIGE[1], KAZUKI KATO[1], NAOKI KURIHARA[2], KATSUMI ONO[3], AND MASAKI HASHIMOTO[1], (Member, IEEE)

[1]Institute of Information Security, Yokohama, Kanagawa 221-0835, Japan
[2]Ernst & Young ShinNihon LLC, Chiyoda, Tokyo 100-0006, Japan
[3]Japan Cybercrime Control Center, Chiyoda, Tokyo 101-0052, Japan

Corresponding author: Masaki Hashimoto (hashimoto@iisec.ac.jp)

ABSTRACT In recent years, the number of fake shopping sites that scam people out of their money or steal their personal information has skyrocketed. To address this problem, Japanese law enforcement agencies such as the police have been detecting fake shopping sites through information provided by a third party and by conducting manual investigations. However, this current approach is quite inefficient. Despite a number of recent studies that use machine learning to detect fake sites, there is still no system for automatically detecting fake shopping sites. Therefore, in this study, we developed an automatic detection system for fake shopping sites to solve the problem of detection inefficiency faced by law enforcement agencies in Japan. The proposed system successfully identified an average of 118,000 target URLs per day from the list of newly registered domains and collected an average of 51,000 sets of HTML data. Also, it was able to determine with 98.5% accuracy using machine learning whether the collected data were fake shopping sites or not. Since this system was able to meet the time requirements for actual operation, we developed an automatic detection system for fake Japanese shopping sites.

INDEX TERMS Domain, fake website, machine learning, URL, web crawling.

I. INTRODUCTION

In recent years, the widespread use of the Internet has led to an increase in crimes in cyberspace. Japan Cybercrime Control Center (JC3) and Anti-Phishing Working Group (APWG), an international non-profit anti-phishing organization, published a joint report [1] in 2018. The report analyzes criminals operating fake shopping websites by classifying them into six groups, and explains that one group, which used to target Japan, has shifted its target to the rest of the world during the course of the analysis. This suggests that the threat of fake shopping sites is spreading from Japan to other countries in the world. In fact, for example, the November 2022 report of the American Association of Retired Persons (AARP) [2], an American non-profit organization (NPO), states: a majority (76%) of Americans age 18 and older have experienced some form of fraud, specifically when purchasing products through online advertisements or fraud related to fake travel reservations.

Additionally, an increasing number of victims have been directed to fake shopping sites from web search results. According to JC3, the number of reported fake shopping sites shared to JC3 by Safer Internet Association [3] in 2021 was 17,878. This is an increase of 7,783 cases (about 77.1%) compared to 10,095 cases in 2020 (Fig. 1) [4].

Under these circumstances, law enforcement agencies in Japan are attempting to crack down on them. However, at present, the majority of opportunities to identify fake shopping sites are provided by consultations with victims and visual inspection by investigators. As a result, only very limited progress has been made despite the significant costs involved in identifying fake shopping sites. The detection of fake shopping sites by Japanese law enforcement agencies is problematic in terms of work efficiency.

On the other hand, numerous studies have utilized machine learning for fake website detection. For example, there was

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

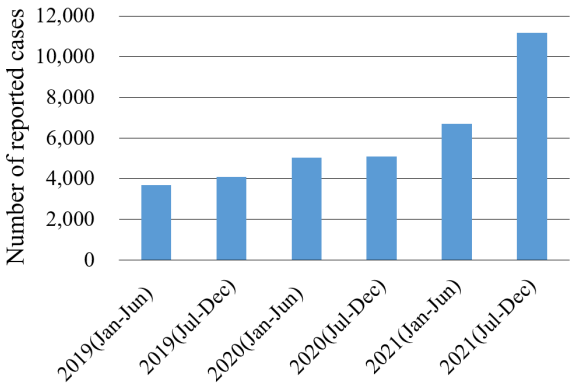


FIGURE 1. Number of reported cases of fake shopping websites. (Shared to JC3 by safer internet association).

a recent study that focused on domain names and utilized LightGBM to determine fake websites [5]. Also, the previous study of this research utilized machine learning to determine fake shopping sites [6]. Although these existing studies reported high detection accuracy, they only aimed to improve accuracy and did not take into account the time requirement (speed) to perform as a practical system for detecting fake websites.

Therefore, the contributions of this research are as follows:

- A proposal of a method for detecting fake shopping sites that satisfies the time requirement to serve as a practical system.
- Automation of the entire detection process from collection to determination of fake shopping sites to solve the inefficiency problem in Japanese law enforcement agencies.

Note that the fake shopping websites considered in this study are a type of fake websites.

In the following sections of this paper, we first describe related research in Section II. Then we describe the design and implementation of the proposed system in Section III, and we explain the evaluation results in Section IV. Finally, we summarize the paper and describes future challenges in Section V.

II. RELATED STUDIES AND THIS STUDY

A. STUDIES ON METHODS FOR IDENTIFYING FAKE WEBSITES

Recent studies utilize machine learning for fake website detection. These existing studies can be broadly categorized into the following four groups:

- Studies focusing on strings in URLs and domain names
- Studies focusing on the tree structure of HTML
- Studies focusing on HTML source code
- Other studies

Machine learning-based fake site detection methods are generally assessed on the basis of detection accuracy. Most of the recently proposed methods have achieved an accuracy rate of over 90%, hence it can be said that the effectiveness of machine learning for detecting fake sites has been proven. In the following, we describe the existing studies for each category.

1) STUDIES FOCUSING ON STRINGS IN URLS AND DOMAIN NAMES

Extensive studies classify and identify fake sites using URL and domain name strings. For example, there is a study that focuses on domain names and uses LightGBM to identify fake sites [5], and a study that uses a specific algorithm to evaluate URL strings to identify fake sites [7]. There is also a study on machine learning of domain characteristics observed in attacks using fast-flux, double-flux, domain-flux (DGA), for use in identifying fake sites [8], and a study on machine learning focusing on URL strings of pages with login forms to identify fake sites [9]. Many other related studies have also been published [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. As mentioned earlier, all of these proposals demonstrated that they were able to detect fake sites with high accuracy.

2) STUDIES FOCUSING ON THE HTML TREE STRUCTURE

Examples of studies that classify and identify fake sites from the HTML tree structure include the following; a study that attempts to detect fake sites using machine learning by collecting HTML Document Object Model (DOM) information limited to the top sites based on Alexa rankings [21], a study that attempts to identify fake sites by applying a Support Vector Machine (SVM) learning model to datasets related to Rakuten, Japan's leading shopping site [22], a study to identify fake sites by machine learning applying a kernel method to efficiently analyze DOM tree structure [23], and a study on machine learning to identify fake sites by combining the structure of the DOM with the structure of Cascading Style Sheets (CSS) [24]. All of these studies utilizing tree structures and machine learning have also reported highly accurate detection of fake sites.

3) STUDIES FOCUSING ON HTML SOURCE CODE

The following are examples of studies that classify and identify fake websites from HTML source code. First, there is a study on the identification of fake shopping sites using machine learning, focusing on fake shopping sites among fake sites [6]. There is also a study on machine learning to identify fake sites, focusing on hyperlinks in source code [25]. In addition, there is a study to identify fake sites by attempting machine learning on Java script, source code, page content, and style using 6321 legitimate datasets on COVID-19 from WhoisDS and an equal number of fake site datasets from the domain tools dataset [26].

4) OTHER STUDIES

Examples of studies that utilize external systems (e.g. search engines) to identify fake sites include the following; a study that uses a combination of incoming emails and a search

engine to identify fake sites [27], and a study that utilizes Microsoft Reputation Services (MRS) to classify URLs to identify fake sites [28].

Some studies have also combined the previously mentioned approaches to identify fake sites. The followings are examples of this. Firstly, there is a study attempting to identify fake sites by URL information, domain age, subdomains, anchor URLs, IP addresses, and so on [29]. There is also a study focusing on URL information and DNS-related information associated with the domains [30]. In addition, there is a study that references 6 sites including Alexa as correct sites and 6 sites including phishing sites as fake sites, and uses a dataset of 3,980,870 URLs in total [31]. This study attempts machine learning with a combination of the perspectives including linguistic features, human-engineered features, deep-web features, URL segmentation, host-based features, and content-based features.

B. STUDIES ON WEB CRAWLING

Various studies on the fake site identification method explained previously in II-A are all aimed at improving the identification accuracy using some pre-prepared research datasets. However, when designing a realistic automatic fake shopping site detection system, the ability to collect the data to be examined is necessary. Therefore, in this subsection, we describe existing research on web crawling.

Olston et al. broadly classified web crawling by operation into incremental crawling and batch crawling. They classified most commercial crawlers as incremental crawling, and claimed that the main challenges of web crawling are the coverage and newness of the information [32]. In Castillo's research, it is assumed that the existing research on web crawling will be used as an internal function of a search engine, and he raised the issue that there has not been much progress in terms of collecting research data [33]. Tchakounte et al. attempted to collect fake website information by starting from a search query, selecting URLs, exploring URLs and detecting duplicates, extracting and pre-processing page content, evaluating similarity, storing, and so on. They claimed to be the first crawler designed for fake sites [34]. According to the research by Batsakis et al., an important evaluation index for web crawlers is the harvest rate [35].

C. SUMMARY OF RELATED STUDIES AND ABOUT THIS STUDY

The aim of this study is to develop a system that detects the URLs of fake shopping sites through data collection by a web crawler and machine learning focusing on HTML source code. In terms of a web crawler for fake sites, it is similar to the work of Batsakis et al. As a detection method, it can be classified as one based on machine learning on HTML source code, as described in II-A3. The reason for choosing the machine learning method on source code is that, especially for fake shopping sites (unlike phishing sites such as fake login sites that aim to steal IDs and passwords), it is difficult to distinguish unless looking at the source code itself, not the URL or the tree structure of the site.

In addition, as mentioned earlier, most existing studies use pre-prepared datasets to improve identification accuracy of machine learning. Hence, we believe that there is insufficient research and consideration for the development of practical, all-round systems that collect and identify data from actual websites that are open to the public. In other words, in order to develop an automatic detection system for fake shopping sites, the ability to collect HTML source code for evaluation is essential, in addition to the ability to judge authenticity through machine learning. This requires accessing the websites and actually rendering them, which is a work-intensive task. Therefore, it is particularly important to develop an effective web crawling function in the development of an automatic detection system for fake shopping sites.

Considering these factors, the main contributions of this study are summarized in the following three points.

1) Designing and implementing an efficient web crawler for fake shopping sites.
2) Evaluating various machine learning-based identification methods by applying them to real-world websites, selecting a method suitable for detecting fake shopping sites, and implementing the method.
3) Combining the above two points as one system.

III. DESIGN OF AN AUTOMATIC DETECTION SYSTEM FOR FAKE SHOPPING SITES

A. OVERVIEW OF THE AUTOMATIC DETECTION SYSTEM FOR FAKE SHOPPING SITES

The system we have developed consists of a Web Crawling Subsystem (WCS) and a Machine Learning Subsystem (MLS). The WCS automatically crawls and downloads the HTML source code, and the MLS reads the data and determines whether it is a fake shopping site or not. This system connects these functions and is a comprehensive system that aims to automatically realize everything from crawling to judging whether a website is a fake shopping site or not, targeting new websites that are launched daily in the Internet space (Fig. 2).

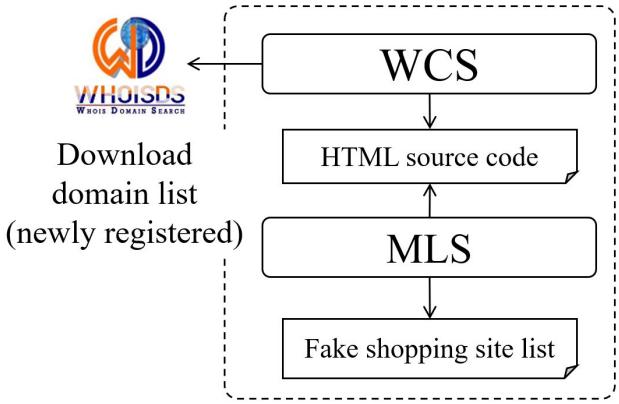


FIGURE 2. System overview.

The process flow of our system is that the WCS downloads domains two days after they are newly registered, collects the HTML source code based on the list generated from the data, and the MLS reads the HTML source code to identify fake shopping sites.

Our system requires the entire process to be completed within 24 hours, as newly registered domains are updated on a daily basis. This is to ensure that the conditions can be met even after five years, when the number of newly registered domains is on the rise, and we set the time requirement to complete the entire process within 6 hours (WCS within 4 hours on average, MLS within 2 hours on average). We also set the target WCS's harvest rate (collection rate) of more than 20%, as shown in the existing study [36] where data were presented together with speed, and the target MLS's judgment accuracy of more than 88.3%, as shown in the existing study [6]. The upward trend in the number of newly registered domains was estimated based on 30 datasets on the number of new domains that have been acquired between August 1st and September 9th, 2022 (Fig. 3) as a preliminary experiment. Also, we consider the useful life of the software at five years, the same as the amortization period under the Japanese tax law.

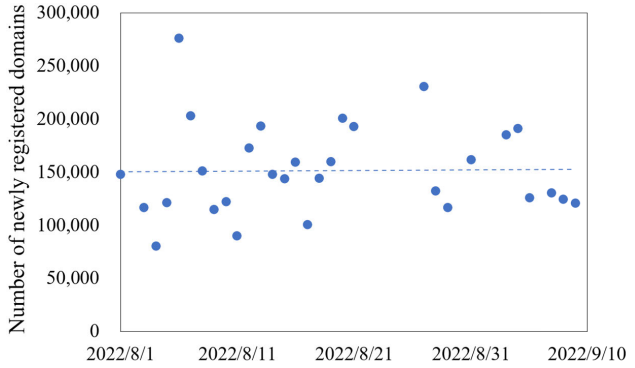


FIGURE 3. Trends in the number of newly registered domains. (Preliminary experiment results).

Our system is activated once a day at 2:00 am by a CRON batch. The WCS downloads domains from WhoisDS two days after they are newly registered, and downloads the HTML source code of the website launched for each domain.

Among the websites of newly registered domains collected here, we look for newly set up fake shopping sites. However, there are tens of thousands of new websites that can be accessed every day, and most of them are legitimate websites that do not fall under the category of fake shopping sites. In reality, it is difficult to visually investigate each site with the human eye and determine whether it is a fake shopping site, because it requires a huge amount of manpower and time.

Therefore, our system uses machine learning to automate the part of the process that determines whether a site is a fake shopping site or not. The MLS developed in this research reads HTML source code and automatically determines whether it is a fake shopping site or not based on the characteristics of the source code. Note that the purpose of this system is to detect fake shopping sites targeting Japanese consumers, we only target websites that contain Japanese hiragana and katakana characters.

B. THE DESIGN OF THE WCS

The WCS has the ability to download large amounts of HTML source code at high speed. At the same time this subsystem needs to address the following requirements:

- To be able to process at high speed even with a single virtual machine, for portability of the assembled system.
- To not overload the crawled website, and do not violate the laws or etiquette.

The problem with processing speed was that in our previous study [37], it took about one month to crawl one day's worth of data for newly registered domains. The response latency on the website side was found to be a major bottleneck. We solved this issue by developing a general-purpose common module that is able to execute parallel processing with an appropriate level of parallelism at each step until the website returns the HTML source code (Fig. 4).

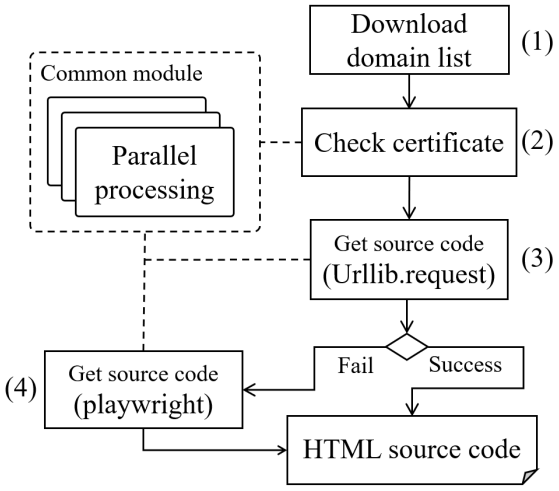


FIGURE 4. Overview of our web crawling subsystem.

The concrete process description is as follows:

1) Get information on the newly registered domains from WHOISDS.
2) (Parallel processing) Check the certificate associated with the newly registered domain and extract the subdomains from the common name of the certificate. Create a list of target sites to retrieve source code excluding port 80 if the access destination is common to port numbers 80 and 443.
3) (Parallel processing) Get HTML source code using Urllib.request.

4) (Parallel processing) Get HTML source code using playwright if Urllib.request failed for reasons such as detection avoidance. (can be skipped in settings)

This subsystem implements all of the above, and is designed to avoid excessive load (repeated or unintended accesses) on the crawled websites and is not in violation of any laws or etiquettes. We are also considering releasing the general-purpose common parallel processing module, which is the core module of this subsystem, as open source in the future.

C. THE DESIGN OF THE MLS

In designing the MLS, previous research [6] used Doc2Vec for vectorization of HTML source code and SVM for the machine learning algorithm to achieve a certain speed and accuracy.

However, the number of newly registered domain websites collected by WCS is expected to reach several hundred thousand per day. There was concern that the processing speed of the system developed in the previous study was not sufficient to handle several hundred thousand websites per day. Another concern is that a large number of false positives would increase the burden of subsequent confirmation work.

Therefore, we designed the MLS with the following requirements:

- Realization of highly accurate judgments that minimize the occurrence of false positives and false negatives.
- Realization of high-speed processing for judging a large number of websites in a short period of time.

1) PREPARATION OF TRAINING DATA

In order to perform classification by machine learning, we first need training data to be trained by a machine learning algorithm. For the training data, we prepared HTML source code for the top pages of about 10,000 fake shopping sites collected by the National Police Agency. We also collected HTML source code of about 10,000 legitimate sites such as shopping sites by ourselves from the Internet. Then we checked all of the collected training data for errors and empty files as part of the cleansing process.

2) SELECTION OF VECTORIZATION METHODS AND MACHINE LEARNING ALGORITHMS

Seeking to improve from our previous study, we decided to select a different vectorization method of HTML source code and a different machine learning algorithm. As for the vectorization method of HTML source code, we chose fastText [38] in this study, whereas Doc2Vec was used in the previous study.

FastText is an open source library developed by Facebook and used for text classification and word embedding for natural language processing. It is a method for representing words and documents as numeric vectors, which can be fed into machine learning algorithms for classification and prediction of text data. FastText can be used to vectorize documents as well as words. We selected fastText because of its high speed in learning and vectorizing documents [39].

As for the selection of a machine learning algorithm for classifying vectorized documents, we decided to use LightGBM(Light Gradient Boosting Machine) [40] in this study, whereas SVM was used in the previous study [6]. LightGBM is a machine learning algorithm based on gradient boosting developed by Microsoft, and is characterized by fast learning and high prediction performance. When selecting a machine learning algorithm, the accuracy of each machine learning algorithm should be compared by training it on the source code of many websites and actually classifying them. However, this would require a great deal of effort and time if done manually. Therefore, in this study, we used PyCaret [41], which is AutoML, for comparison. PyCaret enables comprehensive and automatic comparisons among many machine learning algorithms, and also allows for data pre-processing, parameter tuning, and evaluation with simple programming. A comparison of the accuracy and speed of machine learning algorithms using PyCaret is shown as table (Table 1). The results show that LightGBM is the best in Accuracy, AUC, F1, Kappa, and MCC. Thus, overall, we judged LightGBM to be the most accurate algorithm for classifying whether a site is a fake shopping site or not. Although TT(Sec) was not the top performer, it was not significantly inferior to the other machine learning algorithms. We believe that this can be offset by designing the overall MLS program structure to optimize performance.

3) OUTLINE DESIGN OF THE MLS

MLS is divided into a training function and a classification function.

The training function is invoked manually by the user from the command line. The timing of invocation is determined by the user, but it is expected that the function will be used about once a month to refine the classification model. The classification function is invoked every day in the middle of the night by the batch process. Fig. 5 shows an overview of the MLS designed in this study.

The specific process description is as follows:

1) (Training) Vectorize the HTML source code of the training data.
2) (Training) Train the vectorized data and output a classification model.
3) (Classification) Vectorize the HTML source code of newly registered domain websites collected by the WCS.
4) (Classification) Refer to the classification model created in training, classify whether the website is a fake shopping site or not, and output a list of fake shopping sites.

TABLE 1. Comparison results of machine learning algorithms.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT(Seconds)
Light Gradient Boosting Machine	0.9936	0.9990	0.9981	0.9867	0.9924	0.9868	0.9869	0.3270
Extra Trees Classifier	0.9932	0.9990	0.9999	0.9841	0.9919	0.9860	0.9861	0.2270
K Neighbors Classifier	0.9922	0.9973	0.9946	0.9869	0.9907	0.9840	0.9840	0.5140
Random Forest Classifier	0.9914	0.9989	0.9996	0.9802	0.9898	0.9823	0.9825	0.8200
Gradient Boosting Classifier	0.9877	0.9984	0.9923	0.9787	0.9854	0.9748	0.9749	4.0950
SVM - Linear Kernel	0.9773	0.0000	0.9895	0.9576	0.9733	0.9536	0.9540	0.0240
Decision Tree Classifier	0.9772	0.9757	0.9665	0.9789	0.9726	0.9531	0.9532	0.2190
Linear Discriminant Analysis	0.9765	0.9958	0.9880	0.9571	0.9723	0.9518	0.9522	0.1390
Ada Boost Classifier	0.9753	0.9968	0.9718	0.9693	0.9705	0.9493	0.9493	0.7420
Ridge Classifier	0.9751	0.0000	0.9878	0.9543	0.9708	0.9492	0.9496	0.0190
Logistic Regression	0.9744	0.9954	0.9805	0.9591	0.9697	0.9475	0.9477	0.2610
Quadratic Discriminant Analysis	0.9719	0.9792	1.0000	0.9371	0.9675	0.9428	0.9444	0.1080
Naive Bayes	0.9439	0.9795	1.0000	0.8819	0.9372	0.8869	0.8927	0.0200

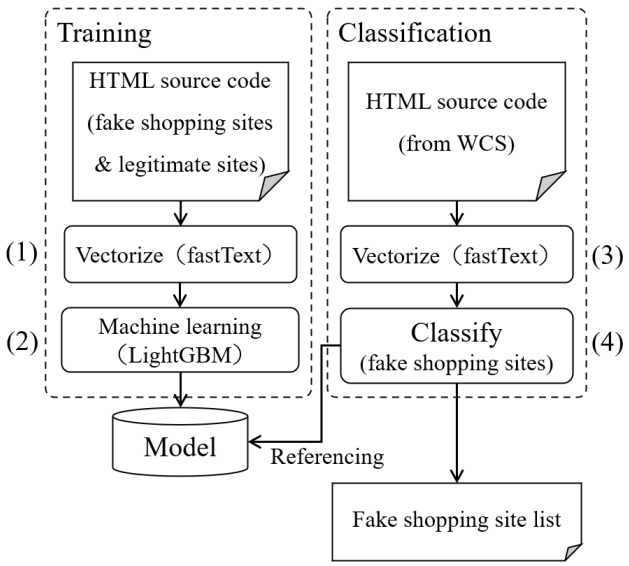


FIGURE 5. Overview of our machine learning subsystem.

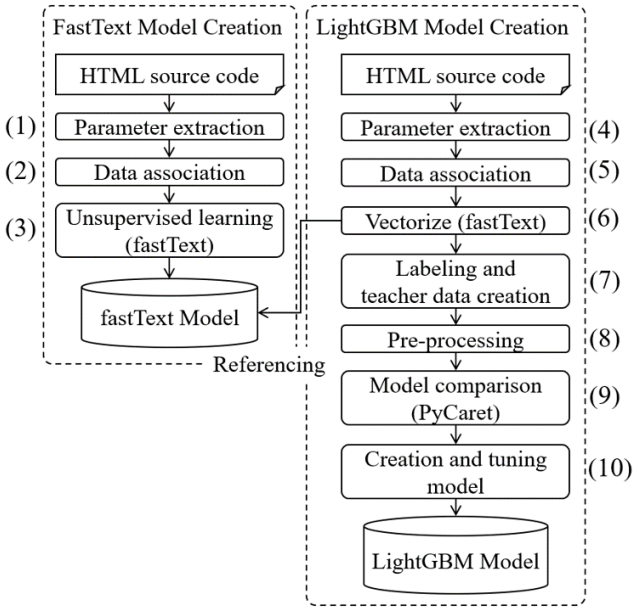


FIGURE 6. Detailed design of the training function.

4) DETAILED DESIGN OF THE TRAINING FUNCTION

The training function consists of a fastText training model creation phase and a LightGBM training model (classification model) creation phase. The specific process is as follows (Fig. 6).

Note that the processes for the fastText model creation phase are prefixed with (fastText) and the LightGBM model creation phase are prefixed with (LightGBM).

1) (fastText) Read the HTML source code of about 20,000 sets of training data in total, extract the src attribute, class attribute, and comments from each HTML source code, and concatenate them into a single document.
2) (fastText) Concatenate all documents created in (1) into a single file and output as processed training data suitable for training with fastText.
3) (fastText) Create a fastText training model by performing unsupervised training on the file created in (2) using fastText's train_unsupervised method.
4) (LightGBM) Read the HTML source code of the training data and extract the src attribute, class attribute, and comments as in (1).
5) (LightGBM) Contatenate the extracted data into a single document.
6) (LightGBM) Refer to the fastText training model created in (3), and vectorize the single document created in (5) with fastText's get_sentence_vector method.
7) (LightGBM) Create teacher data by labeling each piece of data as a fake shopping site or not.
8) (LightGBM) Read in the teacher data and pre-process the data using PyCaret's setup function (automatic processing of missing values, specification of data type, division of training and test data, etc.). The split ratio of training and test data was set at 70% for training data and 30% for test data.

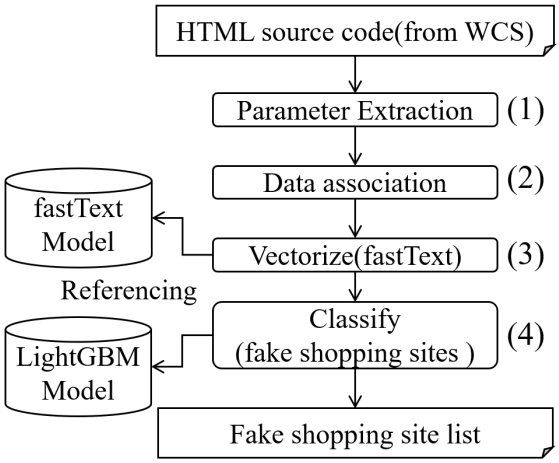


FIGURE 7. Detailed design of the classification function.

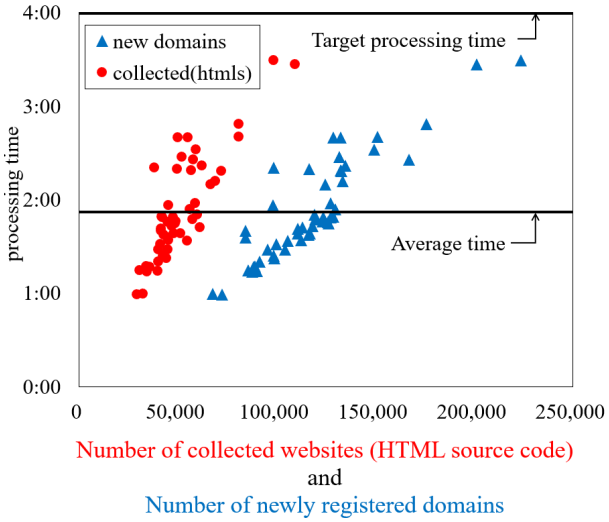


FIGURE 8. Web crawling results.

9) (LightGBM) Train and create machine learning models using PyCaret's compare_model function, and compare multiple machine learning models.
10) (LightGBM) Create a model with PyCaret's create_model ("light_gbm"), tune the hyperparameters of the model with tune_model, and finalize the LightGBM training model with finalize_model.

The reason for extracting the src attribute, class attribute, and comments in (1) and (4) is that previous research [6] has shown that eliminating as much noise as possible from the data improves classification accuracy the most.

Note that Table. 1 is the output in (9), and the optimal machine learning model can be compared from the results.

5) DETAILED DESIGN OF THE CLASSIFICATION FUNCTION

The classification function is performed on all collected HTML source code of newly registered domain sites and lists those determined to be fake shopping sites.

The specific process is as follows (Fig. 7).

1) Read the HTML source code of the website collected by WCS to be classified, and extract the src attribute, class attribute, and comments.
2) Concatenate the extracted data into a single document.
3) Vectorize the extracted data using the fastText's get_sentence_vector method by referring to the fastText training model created by the Training Function described in III-C4.
4) The vectorized website is judged to be a fake shopping site or not by using PyCaret's "predict_model" function with the LightGBM learning model created in III-C4. The "predict_model" function outputs the Label (whether or not it is a fake shopping site) and the Score (the probability that the judgment result is correct).

IV. EVALUATION AND DISCUSSION

A. EVALUATION OF THE WCS

Table 2 and Fig. 8 show the number of domains acquired from WhoisDS, the amount of HTML data acquired, and the time taken to acquire HTML data in the web crawling conducted in this study.

New domains are the number of newly registered domains obtained from WhoisDS, and *collected(htmls)* are the number of newly registered domains for which the HTML source code was obtained.

As a result, the maximum processing time of the WCS was about 3 hours and 30 minutes, the average processing time was 1 hour and 53 minutes, and the data collection rate was about 44%. Therefore, we assess that our targeted performance requirements are sufficiently satisfied.

In addition, we compared the speed of web crawling, using the study by Zowalla et al. [36] as an example, for which data are publicly available (Table 3). Assuming that the environment used by Zowalla et al. has approximately 32 times higher specifications (4 times more CPU cores and 8 times more memory), the crawling speed of this study was multiplied by 32 for comparison. As a result of the comparison, the crawler developed in this study is expected to be about 24 times faster than the web crawling system of Zowalla et al.

B. EVALUATION OF THE MLS

We evaluated the MLS by conducting experiments on two aspects: accuracy and processing time.

For the accuracy evaluation, we prepared 1,000 fake shopping sites' source code that are actually published on the Internet and 1,000 legitimate sites' source code (not fake shopping sites) as experimental data. We fed these data into the MLS to automatically determine if there were any false positives or false negatives. The experiment resulted in only four false positives and 26 false negatives, achieving a high accuracy rate of 98.5%.

TABLE 2. Web crawling results.

date	new domains	target urls	collected	time
2022/10/1	127,245	127,245	47,665	1:48:33
2022/10/2	87,710	87,709	34,482	1:13:53
2022/10/3	85,704	85,703	30,564	1:14:24
2022/10/04	118,062	118,061	46,968	1:42:49
2022/10/05	129,661	129,661	55,993	1:53:39
2022/10/06	132,302	132,301	55,147	2:40:00
2022/10/07	132,548	132,548	56,878	2:18:39
2022/10/11	112,403	112,403	45,395	1:33:58
2022/10/12	126,251	126,251	48,716	1:44:53
2022/10/13	123,491	123,491	42,147	1:48:29
2022/10/14	116,210	116,210	42,735	1:37:18
2022/10/15	166,940	166,940	57,700	2:25:44
2022/10/16	128,682	128,681	41,971	1:48:55
2022/10/17	125,909	125,908	44,949	1:45:42
2022/10/18	175,747	175,747	80,996	2:48:23
2022/10/19	223,537	223,536	98,446	3:29:05
2022/10/20	89,899	89,898	39,867	1:14:05
2022/10/21	91,412	91,411	40,193	1:20:10
2022/10/22	95,601	95,601	40,105	1:27:52
2022/10/23	89,115	89,114	34,137	1:17:20
2022/10/24	72,345	72,345	29,437	0:59:03
2022/10/25	98,676	98,676	42,637	1:23:59
2022/10/26	98,767	98,766	44,293	1:22:34
2022/10/27	104,367	104,367	44,848	1:28:07
2022/10/28	110,854	110,854	48,155	1:38:18
2022/10/30	88,344	88,343	35,900	1:16:36
2022/10/31	110,813	110,813	41,454	1:41:10
2022/11/01	99,876	99,876	41,043	1:31:15
2022/11/02	116,754	116,754	51,351	1:38:18
2022/11/03	116,575	116,575	49,606	2:19:27
2022/11/04	123,170	123,170	49,112	1:45:54
2022/11/05	98,643	98,639	38,068	2:20:22
2022/11/06	149,478	149,478	59,140	2:32:08
2022/11/07	128,695	128,695	50,167	2:39:43
2022/11/08	98,157	98,157	45,197	1:56:20
2022/11/09	134,879	134,879	62,113	2:21:41
2022/11/10	131,895	131,895	52,000	2:27:22
2022/11/11	124,783	124,782	66,667	2:09:30
2022/11/12	84,396	84,396	41,519	1:39:53
2022/11/13	84,295	84,295	44,815	1:35:50
2022/11/16	132,879	132,879	72,103	2:18:13
2022/11/18	127,398	127,398	58,724	1:57:38
2022/11/23	67,761	67,761	32,379	0:59:30
2022/11/24	113,138	113,138	61,198	1:42:05
2022/11/25	119,545	119,545	57,339	1:47:13
2022/11/26	119,368	119,368	59,870	1:50:28
2022/11/27	151,097	151,097	80,853	2:40:15
2022/11/28	133,564	133,563	69,130	2:11:50
2022/11/29	201,279	201,278	109,271	3:26:47
2022/11/30	105,755	105,755	54,686	1:33:44
Average	118,520	118, 519	51,563	1:53:23

TABLE 3. Comparison of web crawling results between other research and this research.

-	The study of Zowalla *et al.*	This study
Virtual machines	22	1
CPU	Intel Xeon E5-2689	Intel Xeon E-2324G
CPU cores	8	2
Memory	256GB	32GB
Harvest speed	7 to 10 doc/seconds	7.58 html/seconds
Harvest rate	19.76%	43.51%

For the processing speed evaluation, we tested how long it actually takes to detect fake shopping sites by using machine learning to determine the source code obtained by the WCS. The results are presented in Table 4 and Fig. 9.

Collected htmls(inputs) are the source code of all the sites downloaded by the WCS when it crawled the sites of newly registered domains, and of course the majority of these are not fake shopping sites. *Detected fake shopping sites* are the number of fake shopping sites that were actually identified and detected as a result of judging the large amount of source code.

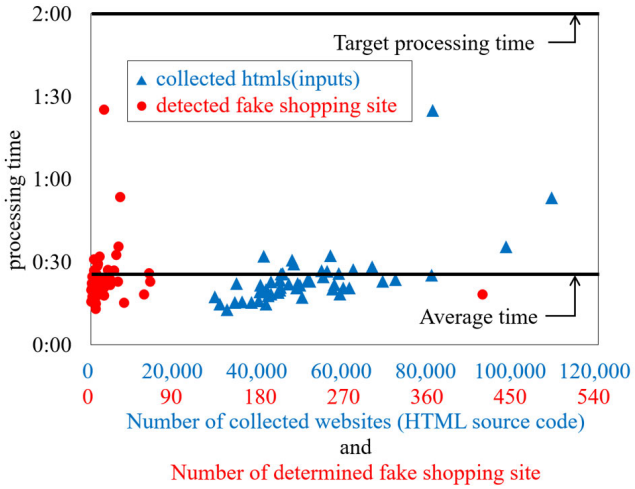


FIGURE 9. Machine learning results.

Our MLS achieved both design targets mentioned in III-A, with an average processing time of 24 minutes compared to our target average processing time of 2 hours, and a judgment accuracy of 98.5% compared to our target judgment accuracy of 88.3%. We are able to confirm that the proposed system is extremely accurate and able to detect fake shopping sites with high processing speed from tens of thousands of newly registered domains per day, making it feasible for practical use.

C. OVERALL EVALUATION

The purpose of this system is to contribute to the elimination and reduction of the threat of fake shopping sites in cyberspace. Therefore, we assessed the effectiveness of this system in terms of these two points; 1) reduction and streamlining of the workload of cyber patrol officers who search for fake shopping sites and provide the information to security service operators and others, and 2) reduction of the man-hours involved in this work.

The URL-based fake site detection method, which is widely used in existing research, has been evaluated as having low accuracy when domain names are divided by hyphens or other delimiters and each part is less than three characters, or when domain names are long without delimiters [6]. We believe that malicious actors will likely be able to easily evade detection by changing URLs or using shortened URLs. Therefore, we excluded the URL-based method from the comparison evaluation.

1) EVALUATION OF REDUCTION AND STREAMLINING OF THE WORKLOAD

In a typical workflow, Japanese cyber patrol officers first use an internet search engine to search for keywords

TABLE 4. Machine learning processing time.

date	htmls	time
2022/10/1	47,665	0:30:50
2022/10/2	34,482	0:22:19
2022/10/3	30,564	0:14:58
2022/10/04	46,968	0:22:18
2022/10/05	55,993	0:26:55
2022/10/06	55,147	0:24:45
2022/10/07	56,878	0:32:29
2022/10/11	45,395	0:25:57
2022/10/12	48,716	0:20:53
2022/10/13	42,147	0:18:01
2022/10/14	42,735	0:22:57
2022/10/15	57,700	0:22:04
2022/10/16	41,971	0:19:21
2022/10/17	44,949	0:20:58
2022/10/18	80,996	1:25:10
2022/10/19	98,446	0:35:41
2022/10/20	39,867	0:16:14
2022/10/21	40,193	0:21:44
2022/10/22	40,105	0:19:26
2022/10/23	34,137	0:15:22
2022/10/24	29,437	0:17:28
2022/10/25	42,637	0:18:28
2022/10/26	44,293	0:19:15
2022/10/27	44,848	0:23:38
2022/10/28	48,155	0:29:19
2022/10/30	35,900	0:15:44
2022/10/31	41,454	0:20:02
2022/11/01	41,043	0:32:10
2022/11/02	51,351	0:23:35
2022/11/03	49,606	0:21:37
2022/11/04	49,112	0:22:24
2022/11/05	38,068	0:15:29
2022/11/06	59,140	0:18:22
2022/11/07	50,167	0:17:13
2022/11/08	45,197	0:25:44
2022/11/09	62,113	0:27:13
2022/11/10	52,000	0:23:04
2022/11/11	66,667	0:28:32
2022/11/12	41,519	0:15:05
2022/11/13	44,815	0:19:59
2022/11/16	72,103	0:23:49
2022/11/18	58,724	0:25:52
2022/11/23	32,379	0:12:56
2022/11/24	61,198	0:20:42
2022/11/25	57,339	0:20:19
2022/11/26	59,870	0:20:52
2022/11/27	80,853	0:25:34
2022/11/28	69,130	0:23:08
2022/11/29	109,271	0:53:28
2022/11/30	54,686	0:27:05
Average	51,563	24:08

frequently used on bogus shopping sites, such as ''cheap'' and ''free shipping''. Then, they manually search the search results using different keywords until they find a link that seems to lead to a fake shopping site. If such a link is found, cyber patrol officers visually check the website and subjectively determine whether or not it matches the common characteristics of fake shopping sites, such as the use of ''suspicious Japanese'', ''common phrases'' and ''abnormally low prices''. Then, after carefully conducting the confirmation and scrutiny process, they compile a list of sites that they have identified as fake shopping sites. Finally, they provide the list to the security services. Therefore, manually searching for fake shopping sites in this way requires a lot of effort and time, and cyber patrol officers need to have a lot of knowledge about fake shopping sites.

In a new workflow using our developed system, instead of using a search engine, our fake shopping site detection system

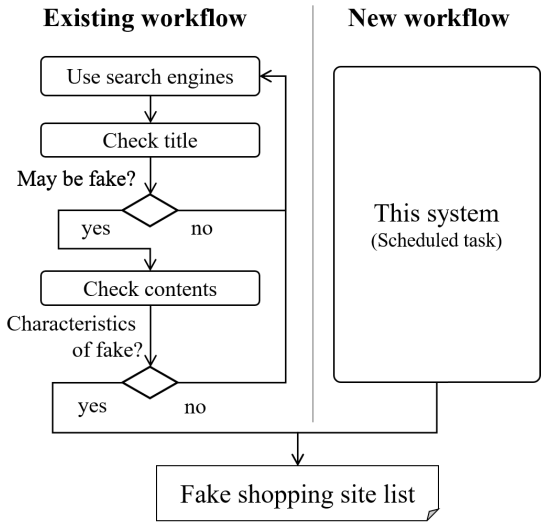


FIGURE 10. Task flow comparison.

automatically outputs a potential list of fake shopping sites through a scheduled process from newly registered domain information. The cyber patrol officers will then only have to check the list.

Therefore, the practical application of this system is expected to significantly reduce the work procedures of cyber patrol officers, leading to a reduction and streamlining of their workload (Fig. 10).

2) EVALUATION OF MAN-HOUR REDUCTION

We surveyed a group of experts involved in cyber patrol operations and cybersecurity volunteers who support that work, regarding the number of man-hours involved.

The man-hours required for cyber patrol activities were calculated by referring to the previous study [42], and excluding the preparation of other input information required in the process of work. The time required for the volunteers was about 1 hour and 40 minutes for cyber patrolling and about 20 minutes for summarizing and reporting the results, totaling about 2 hours. They detected and reported about 10 fake shopping sites. The time required for the experts was about 50 minutes for cyber patrolling and about 20 minutes for summarizing and reporting the results, totaling about 1 hour and 10 minutes. They detected and reported about 20 fake shopping sites. The calculation of man-hours and comparison of the results may vary due to the following concerns:

- Differences in the proficiency level of cyber patrols among measured subjects.
- Change in the number of publicly available fake shopping sites at the time of measurement.
- Performance/specifications of the computer used by the measured subject.

However, the new workflow using our developed system will remove the time required for cyber patrolling, which is the

TABLE 5. Work efficiency evaluation.

Item	Volunteer	Expert	This study
Cyber patrol	100min.	50min.	0min.
Summarize, report	20min.	20min.	20min.
Detections	10	20	21
Man-days	5.00	2.92	0.83
Detection rate(sites/min.)	0.08	0.29	1.05

most time-consuming part of the work process. Therefore, a significant reduction in man-hours and an improvement in detection efficiency per unit of time is expected (Table. 5).

D. DISCUSSION AND FUTURE CHALLENGES

1) DISCUSSION ON THE WCS

The WCS satisfies the predetermined processing time requirements with plenty of margin, and as of now, no practical problems have arisen. However, if the number of newly registered domains increases more rapidly than our expectation, there is a possibility that the processing will not be completed within the specified time. Specifically, when the number of newly registered domains per day exceeds about one million, the processing cannot be completed, and data acquisition for the remaining newly registered domains is carried over to the next day.

If this were a short-term issue, there would be no problem. However, if the number of newly registered domains becomes more than about 1 million on a regular basis, it will be difficult for the current WCS to handle it. As mentioned earlier, the response latency on the website side is a bottleneck in data acquisition. As a solution to this problem, it will be possible to handle up to 1.25 million newly registered domains per day by adjusting the parallel number with the pre-installed parallel number adjustment function for data acquisition processing.

In addition, further study can be conducted on the timing of data acquisition from each newly registered domain and the identification of URLs for which data is to be acquired. The current WCS is designed to acquire data about two days after the domains are newly registered. Therefore, content posted after the third day cannot be retrieved. Ideally, the data acquisition process should be executed after the content has been placed on each website, but this timing varies widely from website to website and is difficult to uniquely define. It is conceivable that the data acquisition process could be executed after a longer period of time following the new registration of a domain. However, this is also highly problematic, as early action is needed to reduce the number of victims of fake shopping websites. Therefore, we need to determine the appropriate timing for data acquisition, which is neither too early nor too late. For this decision, we need to analyze how long it takes from new registration of the domain to the actual placement of the content, but this is a challenge for the future.

For the identification of the URLs for data acquisition, we infer the subdomains from the common name of the SSL certificate set up on the website. For wildcard SSL certificates that cannot be inferred from the common name, we use the domain name itself as the URL to retrieve the data. Because of this, it is currently not possible to retrieve data associated with subdomains using a wildcard SSL certificate. In addition, since we specify that the URL for data acquisition is the root directory of the website, data cannot currently be acquired if the content is located outside of the root directory. Ideally, it should be possible to accurately identify URLs including subdomains and content paths. Inference of so-called malicious URLs has been studied in various places and we plan to test the validity of inference using machine learning in the future. In addition, we plan to verify the effectiveness of OSINT information such as SHODAN for subdomains and the validity of site information such as *robots.txt* and *sitemap.xml* for content paths.

2) DISCUSSION ON THE MLS

The MLS also satisfies the pre-determined processing time requirements with plenty of margin, and currently no practical problems have arisen. In addition, even if the number of newly registered domains increases more rapidly than expected, we estimate that up to approximately 2 million domains can be processed within a pre-determined timeframe. As mentioned earlier, the daily processing limit of the WCS is about 1.25 million domains, which is less than the limit of MLS. Therefore, we believe that the processing time of the MLS will not be a problem in the near future, since the design of WCS or the entire system configuration needs to be reviewed before the MLS reaches its limit.

The accuracy of the machine learning was 97.4% for recall and 99.6% for precision. We believe that this is sufficiently practical at this point in time in terms of judgment accuracy and other factors. Currently, human verification is performed as a final judgment against false positives detected by the system. Manual checks are made using a separate rule-based system to save labor, and there have been no operational problems even on days with a large number of detection. As for false negatives, the machine learning model used in this study has an accuracy of 0.9936, so the number of false negatives in actual operation is considered to be extremely small, and we have decided to allow for this. We believe that this allowance does not impair the usefulness of the proposed system. However, there is a concern that the accuracy will decline in the future due to the improvement in the proficiency of fake shopping site creators and the use of adversarial AI. As a countermeasure, we plan to feed back misjudged sites to the machine learning model about once a month for retraining.

3) OVERALL DISCUSSION

From the evaluation of reduction and streamlining of the workload and man-hour reduction, we were able to confirm that our developed system is able to detect fake shopping sites effectively and efficiently. As a result, cyber patrol officers

can focus on doing more advanced and essential work such as publicity and awareness activities against fake shopping sites, criminal analysis, and site take down activities.

The ideal form of an Automatic Detection System for Fake Shopping Sites is one that can detect fake shopping sites on the Internet without human intervention. However, since the proposed system only handles newly registered domains, it does not support existing domains, which is a problem in terms of its comprehensiveness. To address this issue, it may be possible to support existing domains by combining the proposed system with Google's CustomSearchAPI, and Meta's ad library search.

In terms of the internal validity of the workload reduction in this research, there is a dependency between the proficiency in identifying fake shopping sites, the time required for cyber patrols and the number of detected fake shopping sites per unit time. We conducted experiments under the hypothesis that there is no dependency in summarizing the results, and obtained data to support that the hypothesis is correct. However, there are some concerns about the internal validity, such as the small number of qualified experts recruited as participants (only 5 people) and the significant variation in cyber patrol experience among the student volunteers. Nevertheless, we believe that this system significantly reduces the time and effort required for cyber patrolling, which is the most time-consuming part of the work process, leading to substantial reductions in time and workload, and an improvement in detection efficiency per unit time.

Regarding the external validity, since this system is specialized for Japanese fake shopping sites, additional investigation is required to determine the possibility of generalizing it to overseas sites due to potential differences in source code characteristics. Additional research is also needed to determine whether this can be used not only for fake shopping sites but also for fake sites in general. Considering that the source code of fake sites contains various characteristics, we would like to work on the generalization of this system as a future work.

V. CONCLUSION

In this study, we were able to dramatically increase the amount of HTML source code that can be obtained from newly registered domains by using the web crawling subsystem compared to previous studies. At the same time, we succeeded in determining fake shopping sites from a large amount of HTML source code with high speed and accuracy by using the machine learning subsystem. By combining these two subsystems, we constructed a comprehensive system that can perform everything from automatic crawling of the Internet space to automatic determination of fake shopping sites. This system has streamlined the process of searching for and judging fake shopping sites and has significantly reduced the man-hours involved. We have succeeded in developing a system that is sufficiently practical to contribute to the reduction and elimination of the threat of fake shopping sites in cyberspace. This system has been adopted as part of the countermeasure system for fake shopping sites and is currently in operation at JC3.

As for future extension, as the Better Business Bureau cautions against extremely cheap products [43], we believe that the characteristics of fake shopping sites are similar in foreign languages. Therefore, we expect that the knowledge obtained from this research can be applied to countermeasures against fake shopping sites in other countries as well. In addition, we would like to expand the detection targets of this system to include other malicious websites such as phishing sites and technical support fraud sites, and aim to reduce threats and victims in cyberspace by detecting malicious sites more widely and swiftly.

REFERENCES

[1] Japan Cybercrime Control Center (JC3) and Anti-Phishing Working Group (APWG). (2018). *Revealed Threat of Fake Store*. [Online]. Available: https://www.jc3.or.jp/threats/upload/threats_JC3_APWG_Revealed_Threat_of_Fake_Store.pdf

[2] AARP Research. (2022). *Preparing for the Holidays? So Are Criminals Already-Rampant Fraud Expected to Spike*. [Online]. Available: https://www.aarp.org/content/dam/aarp/research/surveys_statistics/econ/2022/holiday-shopping-scams-survey-us-consumers-2022.doi.10.26419-2Fres.00582.001.pdf

[3] *Safer Internet Association*. Accessed: Apr. 6, 2023. [Online]. Available: https://www.saferinternet.or.jp

[4] Japan Cybercrime Control Center. (2021). *Statistical Information on Malicious Shopping Sites*. [Online]. Available: https://www.jc3.or.jp/threats/topics/article-431.html

[5] J. Zhou, H. Cui, X. Li, W. Yang, and X. Wu, "A novel phishing website detection model based on LightGBM and domain name features," *Symmetry*, vol. 15, no. 1, p. 180, Jan. 2023, doi: 10.3390/sym15010180.

[6] N. Kurihara, H. Tsuji, and M. Hashimoto, "Spoofed website detection using machine learning," *IEICE Tech. Rep.*, vol. 118, no. 315, pp. 19–24, Nov. 2018.

[7] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing URL detection using association rule mining," *Hum.-Centric Comput. Inf. Sci.*, vol. 6, no. 1, Dec. 2016, Art. no. 64, doi: 10.1186/s13673-016-0064-3.

[8] E. Kidmose, M. Stevanovic, and J. M. Pedersen, "Detection of malicious domains through lexical analysis," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Glasgow, U.K., Jun. 2018, pp. 1–5, doi: 10.1109/CyberSecPODS.2018.8560665.

[9] M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki, and V. González-Castro, "Phishing URL detection: A real-case scenario through login URLs," *IEEE Access*, vol. 10, pp. 42949–42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[10] M. Khonji, Y. Iraqi, and A. Jones, "Lexical URL analysis for discriminating phishing and legitimate websites," in *Proc. 8th Annu. Collaboration, Electron. Messaging, Anti-Abuse Spam Conf.* New York, NY, USA: Association for Computing Machinery, Sep. 2011, pp. 109–115, doi: 10.1145/2030376.2030389.

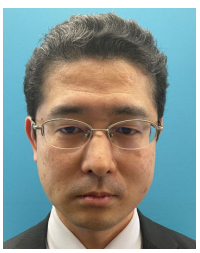
[11] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in *Proc. 3rd ACM Int. Workshop Secur. Privacy Anal.* New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 55–63, doi: 10.1145/3041008.3041016.

[12] V. Vundavalli, F. Barsha, M. Masum, H. Shahriar, and H. Haddad, "Malicious URL detection using supervised machine learning techniques," in *Proc. 13th Int. Conf. Secur. Inf. Netw.* New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 1–6, doi: 10.1145/3433174.3433592.

[13] B. J. Shantanu and R. J. A. Kumar, "Malicious URL detection: A comparative study," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Coimbatore, India, Mar. 2021, pp. 1147–1151, doi: 10.1109/ICAIS50930.2021.9396014.

[14] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical feature based phishing URL detection using online learning," in *Proc. 3rd ACM Workshop Artif. Intell. Secur.* New York, NY, USA: Association for Computing Machinery, Oct. 2010, pp. 54–60, doi: 10.1145/1866423.1866434.
[15] R. B. Basnet, A. H. Sung, and Q. Liu, "Feature selection for improved phishing detection," in *Proc. 25th Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst., Adv. Res. Appl. Artif. Intell. (IEA/AIE)*. Berlin, Germany: Springer-Verlag, 2012, pp. 252–261, doi: 10.1007/978-3-642-31087-4_27.
[16] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. With Appl.*, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.
[17] T. T. T. Pham, V. N. Hoang, and T. N. Ha, "Exploring efficiency of character-level convolution neuron network and long short term memory on malicious URL detection," in *Proc. 7th Int. Conf. Netw., Commun. Comput.* New York, NY, USA: Association for Computing Machinery, Dec. 2018, pp. 82–86, doi: 10.1145/3301326.3301336.
[18] N. Al-Milli and B. H. Hammo, "A convolutional neural network model to detect illegitimate URLs," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 220–225, doi: 10.1109/ICICS49469.2020.239536.
[19] B. Geyik, K. Erensoy, and E. Kocyigit, "Detection of phishing websites from URLs by using classification techniques on WEKA," in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, Jan. 2021, pp. 120–125, doi: 10.1109/ICICT50816.2021.9358642.
[20] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A.-H. Baddar, "URL phishing detection using machine learning techniques based on URLs lexical analysis," in *Proc. 12th Int. Conf. Inf. Commun. Syst. (ICICS)*, Valencia, Spain, May 2021, pp. 147–152, doi: 10.1109/ICICS52457.2021.9464539.
[21] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 1–28, Sep. 2011, doi: 10.1145/2019599.2019606.
[22] J.-C. Xu, K. Shin, and Y.-L. Liu, "Detecting fake sites based on HTML structure analysis," in *Proc. 6th Int. Conf. Commun. Netw. Secur. (ICCNS)*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 86–90, doi: 10.1145/3017971.3017980.
[23] T. Ishikawa, Y.-L. Liu, D. L. Shepard, and K. Shin, "Machine learning for tree structures in fake site detection," in *Proc. 15th Int. Conf. Availability, Rel. Secur.* New York, NY, USA: Association for Computing Machinery, Aug. 2020, pp. 1–10, doi: 10.1145/3407023.3407035.
[24] J. Feng, Y. Qiao, O. Ye, and Y. Zhang, "Detecting phishing webpages via homology analysis of webpage structure," *PeerJ Comput. Sci.*, vol. 8, p. e868, Feb. 2022, doi: 10.7717/peerj-cs.868.
[25] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 2015–2028, May 2019, doi: 10.1007/s12652-018-0798-z.
[26] S. R. Zahra, M. A. Chishti, A. I. Baba, and F. Wu, "Detecting COVID-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining based intelligence system," *Egyptian Informat. J.*, vol. 23, no. 2, pp. 197–214, Jul. 2022, doi: 10.1016/j.eij.2021.12.003.
[27] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Doha, Qatar, Apr. 2008, pp. 840–843, doi: 10.1109/AICCSA.2008.4493625.
[28] M. N. Feroz and S. Mengel, "Phishing URL detection using URL ranking," in *Proc. IEEE Int. Congr. Big Data*, New York, NY, USA, Jul. 2015, pp. 635–638, doi: 10.1109/BigDataCongress.2015.97.
[29] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Inf. Secur.*, vol. 8, no. 3, pp. 153–160, May 2014, doi: 10.1049/iet-ifs.2013.0202.
[30] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2019, doi: 10.1145/3191329.
[31] E. Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An adversarial attack analysis on malicious advertisement URL detection framework," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1332–1344, Jun. 2023, doi: 10.1109/TNSM.2022.3225217.
[32] C. Olston and M. Najork, "Web crawling," *Found. Trends Inf. Retr.*, vol. 4, no. 3, pp. 175–246, Mar. 2010, doi: 10.1561/1500000017.
[33] C. Castillo, "Effective web crawling," *SIGIR Forum*, vol. 39, no. 1, pp. 55–56, Jun. 2005, doi: 10.1145/1067268.1067287.
[34] F. Tchakounte, J. C. T. Ngnintedem, I. Damakoa, F. Ahmadou, and F. A. K. Fotso, "Crawl-shing: A focused crawler for fetching phishing contents based on graph isomorphism," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8888–8898, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.003.
[35] S. Batsakis, E. G. M. Petrakis, and E. Milios, "Improving the performance of focused Web crawlers," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 1001–1013, Oct. 2009, doi: 10.1016/j.datak.2009.04.002.
[36] R. Zowalla, T. Wetter, and D. Pfeifer, "Crawling the German health web: Exploratory study and graph analysis," *J. Med. Internet Res.*, vol. 22, no. 7, Jul. 2020, Art. no. e17853, doi: 10.2196/17853.
[37] K. Kato, "Development of a web crawling system for detecting fake sites," M.S. thesis, Inst. Inf. Secur., Yokohama, Japan, 2021.
[38] *fastText*. Accessed: Apr. 6, 2023. [Online]. Available: https://fasttext.cc/
[39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, *arXiv:1607.01759*, doi: 10.48550/arXiv.1607.01759.
[40] *Welcome to LightGBM's Documentation!* Accessed: Apr. 6, 2023. [Online]. Available: https://lightgbm.readthedocs.io/
[41] *Welcome to PyCaret—PyCaret Official*. Accessed: Apr. 6, 2023. [Online]. Available: https://pycaret.org/
[42] L. Greenberg, "Data available for the measurement of output per man-hour," in *Output, Input, and Productivity Measurement*. Princeton Univ. Press, 1961, pp. 147–200.
[43] Better Business Bureau. (2022). *BBB Tip: Smart Shopping Online*. [Online]. Available: https://www.bbb.org/all/scamstudies/counterfeit_goods_scams/counterfeit_goods_scams_full_studyv

KEISUKE SAKAI received the master's degree in physics from the Tokyo University of Science, Tokyo, Japan, in 2010, and the master's degree in informatics from the Institute of Information Security, Kanagawa, Japan, in 2023. Since 2010, he has been an engineer managing and operating the in-house infrastructure at an IT subsidiary of a fine chemical manufacturer for eight years. Since 2018, he has also been engaged in technical support and research work related to information security at a government agency. Since 2023, he has also been a Visiting Researcher with the Hashimoto Laboratory, Institute of Information Security. His research interests include cybersecurity and OSINT.



KOSUKE TAKESHIGE worked for seven years in the private sector as a software engineer, prior to joining the Police Department. Since 2010, he has been a cybercrime investigator for a police agency. After being dispatched to the Japan Cybercrime Countermeasures Center (JC3), he is currently in-charge of investigating cybercrimes, while also belonging to the Hashimoto Laboratory, Institute of Information Security, as a Visiting Researcher. His main research interests include cybersecurity, software engineering, and artificial intelligence.

KAZUKI KATO received the master's degree in informatics from the Institute of Information Security, Kanagawa, Japan, in 2022. Since 2008, he has been with a telecommunication company and other companies for eight years in the field of network and security. Since 2016, he has also been engaged in technical support and research work related to information security at a government agency. Since 2022, he has also been a Visiting Researcher with the Hashimoto Laboratory, Institute of Information Security. His research interests include cybersecurity and network security.

NAOKI KURIHARA was an infrastructure engineer, from 2009 to 2014, mainly in the design, construction, and operation of servers and networks. He was a police officer, from 2014 to 2020. Predominantly with the Cybercrime Countermeasures Division, he has experienced technical support, such as digital forensics, cybercrime investigation support, and research and development of cybercrime countermeasure systems using machine learning. He joined Ernst & Young ShinNihon LLC, in 2020.

KATSUMI ONO has been with the Japan Cybercrime Control Center (JC3), since March 2021, as the Director of Economic and Financial Crime Countermeasures after working at a government office. At JC3, he is in-charge of industry-academia-government collaboration, including coordination with member companies and related organizations and public relations and awareness-raising work related to cybercrime countermeasures.

MASAKI HASHIMOTO (Member, IEEE) received the Ph.D. degree in informatics from the Institute of Information Security, Kanagawa, Japan, in 2010. Since 2010, he has been an assistant professor. Since 2014, he has been an Associate Professor with the Institute of Information Security. From April 2014 to March 2015, he was a Visiting Researcher with the Information Security Group at Royal Holloway, University of London, U.K. His research interests include intrusion detection/prevention, malware analysis, and OSINT technology.

He is a member of the Information Processing Society of Japan, the Institute of Electronics, Information and Communication Engineers, and the Japan Society for Software Science and Technology. He is also engaged as a member of the Editorial Board of *IEICE/English Journal D*, Ministry of Economy, Trade and Industry/Electrical Safety System WG Committee, and NETSAP2022 Workshop Organizer.