IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 November 2022, accepted 18 November 2022, date of publication 24 November 2022,
date of current version 30 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3224781

RESEARCH ARTICLE

Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection

YI WEI[1] AND YUJI SEKIYA[2]

[1]Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan
[2]Security Informatics Education and Research Center, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan

Corresponding authors: Yi Wei (weiyi@g.ecc.u-tokyo.ac.jp) and Yuji Sekiya (sekiya@si.u-tokyo.ac.jp)

ABSTRACT Phishing is a kind of worldwide spread cybercrime that uses disguised websites to trick users into downloading malware or providing personally sensitive information to attackers. With the rapid development of artificial intelligence, more and more researchers in the cybersecurity field utilize machine learning and deep learning algorithms to classify phishing websites. In order to compare the performances of various machine learning and deep learning methods, several experiments are conducted in this study. According to the experimental results, ensemble machine learning algorithms stand out among other candidates in both detection accuracy and computational consumption. Furthermore, the ensemble architectures still provide impressive capability when the amount of features decreases sharply in the dataset. Subsequently, the paper discusses the factors why ensemble machine learning methods are more suitable for the binary phishing classification challenge in up-date training and real-time detecting environment, which reflects the sufficiency of ensemble machine learning methods in anti-phishing techniques.

INDEX TERMS Phishing websites detection, machine learning, ensemble learning, deep learning.

I. INTRODUCTION

With the expansion of the Internet and the ubiquity of social media, data breaches have consequently emerged as one of the main concerns in cyber security fields. Most security problems and data breaches are usually caused by malicious criminals. Phishing is a common form of cybercrime when hackers attempt to lure individuals into divulging private information, such as bank account details, credit card number, and even employee login credentials for use in unauthorized access to a specific company. To lure a victim, hackers create fraudulent messages that seem to come from a trustworthy person or entity but actually contain disguised links. Then, they send these fake messages to the targets by email or instant messages. If the victim is tricked by the malicious link, confidential data of him or her will be stolen in this cyber fraud.

Since the coronavirus pandemic, people are ordered to work remotely, Covid-19-themed phishing attacks have spiked. Phishers take advantage of the virus-related fear and anxiety of the public in the wake of the spread of the virus. Emails allegedly providing ways to stop the coronavirus outbreak were the most common kind of phishing emails employed [1]. In order to boost the likelihood of success, phishing attempts that occurred during the pandemic also had distinctive features, for instance, the registration of covid-related domains soared during the first months of the pandemic [2]. Threats on social media continued to escalate, with a 47% increase from Q1 to Q2 2022, according to a recent trends report by the APWG (Anti-Phishing Working Group) [3].

Artificial Intelligence (AI) is an emerging science, which has captured tremendous attention over the past decades. It investigates how to build intelligent machines that can creatively find solutions to problems without human intervention. Machine Learning (ML) is a branch of AI that gives machines the capability to automatically learn and make decisions from experience. As a subset of ML, deep learning (DL) employs neural networks with a structure resembling the human neural system to analyze a wide range of variables. Researchers in the cybersecurity domain have conducted various AI solutions to detect illegal phishing attacks.

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

A typical AI-based phishing detection procedure is shown in FIGURE 1, in which AI techniques can learn and extract features to classify phishing attacks effectively and efficiently. Existing phishing detection methods usually choose ML or DL to detect unknown attacks. Due to its ability to automatically extract features, DL has recently been seen as a promising phishing detection tool [4]. However, our research found that based on some generally recognized phishing websites features [5], conventional ML methods achieve higher accuracy and lower false-positive rate. Besides, DL techniques always suffer from deficiencies in computational constraints and time complexity. This study is intended to indicate the sufficiency of traditional ML algorithms for phishing URLs detection.

In summary, this paper makes the following contributions:

- We evaluated multiple ML algorithms for phishing detection empirically and contrasted their performances.
- We implemented and evaluated a 3-layer fully connected neural network (FCNN) model, an LSTM model, and a CNN model on a dataset.
- We analyzed the performances of ML-based methods and DL-based methods. Moreover, we discussed the sufficiency of ML-based methods for phishing detection and provided suggestions for the phishing feature selection approach.

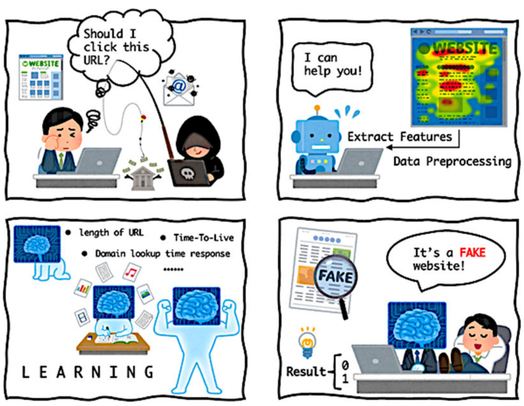

FIGURE 1. Phishing detection steps by applying AI solutions.

The rest of this paper is organized as follows: Section II presents the previous research employing, respectively, ML and DL. Section III introduces and compares three published datasets and features. Section IV provides the detection results by utilizing conventional ML algorithms. In section V, we build several DL models and compare the results with ML. Finally, Section VI discusses ML methods' sufficiency for phishing detection and proposes future works in phishing detection field.

II. LITERATURE REVIEW

Based on the methodologies used, phishing detection solutions can be categorized into many different groups including blacklist and whitelist [6], heuristic-based method [7], visual similarity [8], machine learning, deep learning, and hybrid [9]. This section mainly talks about two categories: ML-based phishing detection techniques and DL-based phishing detection approaches in the literature.

A. ML-BASED PHISHING DETECTION

There are supervised, semi-supervised, unsupervised, and reinforcement methods in Machine Learning, the most popular one used to detect phishing acts is the supervised method, where machines try to make intelligent decisions by learning certain features of phishing and legitimate sample dataset [10]. These kinds of solutions always extract features like URLs [11], [12], [13], hyperlinks information [14], webpage content [15], [16], hybrid features [17], and other resources. The performance of these methods typically depends on the quality of the dataset, the characteristics, and the algorithm employed in the approach [18]. The following are typical ML algorithms used in phishing detection methods: Support Vector Machine, Classification and Regression Tree, Random Forest, AdaBoost, Light Gradient Boosting Machine…etc.

A phishing detection engine using the features extracted from URLs was proposed by A. Butnaru et al. [13]. They also assessed how well phishing detection performed over time without model training. As a result, their solution works better than Google Safe Browsing (GSB), which is the default security tool in most popular web browsers. It is worth mentioning that the model performs well against phishing URLs even after one year. Although the methodology achieves good performance, the authors are concerned about the robustness against adversarial attacks, which are frequently exploited by malevolent entities even when the system produces good performance.

Jain and Gupta [14] presented a novel method that analyzes hyperlinks included in the HTML source code of websites to identify phishing assaults. In their feature selection process, six new features were proposed to increase the detecting performance, which is also the key contribution in this work because both processing time and response time were thus reduced. Moreover, their approach is language-independent to detect any textual language webpage. However, the approach has certain restrictions because it is totally dependent on the website's source code. If the attackers change all the page resource references, their method will make a false prediction.

The performance of an ML-based system heavily depends on the feature sets. Useless features will increase the cost of storage, time, and power. Feature engineering is crucial since traditional ML techniques depend on human expertise for feature extraction and selection. K. L. Chiew et al. [19] introduced a Hybrid Ensemble Feature Selection (HEFS) framework for ML-based phishing detection systems, where major feature subsets are created using a novel Cumulative Distribution Function gradient (CDF-g) method. By using a function perturbation, they can get a set of baseline features. After integrating with Random Forest, the detection accuracy can achieve 94.6% using only 20.8% of the original number of features.

The main agenda of our previous work [20] also focuses on the feature selection approach for phishing detection. In our proposed framework, existing feature importance methods Mean Decrease in Impurity (MDI), Permutation, and SHapley Additive explanation (SHAP) are leveraged to obtain a ranking of the importance of features. By assigning different weights to evaluation metrics under various conditions, we can automatically generate the optimal feature subsets. According to experimental results, our feature selection framework outperforms HEFS [19] on the same dataset. Based on the top 10 features we select, detection accuracy achieves 96.83%, which is higher than their results (94.6%) with 10 baseline features. Both of the feature selection frameworks above can provide a fully automatic, flexible, and robust system to produce high-quality sub-feature sets. Furthermore, the framework can be applied to various datasets, which can provide a solution to the problem discussed in [4] that manual feature engineering is separated from classification tasks in conventional ML models.

B. DL-BASED PHISHING DETECTION

It is precisely because of its capability to find hidden information in complicated datasets, DL has recently emerged as a viable substitute for traditional ML techniques. In order to enhance the effectiveness of phishing detection solutions, various DL-based approaches have been applied. Popular DL algorithms used in phishing detection include Multi-Layer Perceptron (MLP) [21], [22], Long Short-Term Memory (LSTM) [23], Convolutional Neural Network (CNN) [24], [25], Recurrent Neural Network (RNN) [26], [27], and hybrid [28]…etc.

Yerima and Alzaylaee [25] presented a DL-based approach with high detecting accuracy, where CNN is utilized to distinguish legitimate websites from phishing websites. A 1D-CNN model with two convolutional layers, two max-pooling layers, and one fully connected layer was constructed in their method. The model surpassed several popular machine learning classifiers, according to testing on a benchmarked dataset of 4,898 examples from phishing websites and 6,157 instances from reliable websites. However, to fine-tune the important impacting parameters (i.e. number of filters, filter lengths, and the number of fully connected units), they conducted a series of experiments. This time-consuming and labor-intensive procedure is frequently observed in DL-based methods [29], [30].

Li et al. [23] proposed an LSTM-based phishing detection method for big email data which consists of two important stages: sample expansion stage and testing stage. To suit the needs of in-depth learning, sufficient training samples should be provided, they merged KNN with K-Means in the sample expansion stage. Prior to testing, they preprocessed the data by generalizing, word segmenting, and creating word vectors. The LSTM model was then trained using the preprocessed data. Finally, they categorized phishing emails. The accuracy rate of their proposed phishing email detection method can approach 95%, according to experimental results. In their research, to make the detection system more efficient, they labeled a small amount of data manually. Based on this small dataset, they used KNN and K-Means to expand it into the final samples. It is commonly known that DL can manage large amounts of data and when the size of the dataset increases, DL performs better. However, it is difficult for researchers to find abundant and appropriate datasets to work with. At the same time, using a single processor to train DL models on such a significant dataset is also a challenge.

In a recent comprehensive DL-based review in the phishing detection field [4], Do et al. indicated that Each DL algorithm has unique properties that make it ideal for a specific application. For example, RNN is more appropriate for processing sequential data such as natural language and text. When analyzing two-dimensional data, such as images and videos, CNN produces better results. In addition, the main drawback is that supervised DL requires a massive amount of labeled instances, which adds a high level of computational complexity to the detection system [31]. Additionally, DL models are unable to justify the inference they draw. It would be tough to comprehend the relationship between input attributes and output decisions [32].

III. DATASET AND FEATURES

Several high-quality phishing datasets are widely used by various authors in their research, such as UCI_2015 [33], Mendeley_2018 [34], and Mendeley_2020 [35]. Phishing instances are usually derived from PhishTank [36], which is a cooperative repository for data and information about phishing attacks on the Internet. Other legitimate instances are from Alexa, DMOZ, and Common Crawl. Features used in phishing detection are usually extracted from URLs (protocol, domain, path, parameter shown in FIGURE 2) and other external resources. In this section, we will give an introduction and comparison of these three popular phishing datasets.

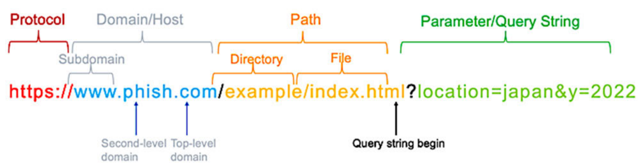

FIGURE 2. An example of URL structure.

A. UCI_2015

University California Irvine Machine Learning Repository (UCI) is a common repository that contains both fraudulent and trustworthy website URLs, which is popular among phishing detection researchers [4], [37], [38]. The dataset was donated in 2015 and collected primarily from PhishTank and MillerSmiles archives. The dataset comprises 30 features and 11055 instances (6157 legitimate websites and

TABLE 1. Features in dataset UCI_2015.

F1	having_IP_Address	F17	Submitting_to_email
F2	URL_Length	F18	Abnormal_URL
F3	Shortning_Service	F19	Redirect
F4	having_At_Symbol	F20	on_mouseover
F5	double_slash_redirecting	F21	RightClick
F6	Prefix_Suffix	F22	Using Pop-up Window
F7	having_Sub_Domain	F23	IFrame Redirection
F8	SSLfinal_State	F24	Age of Domain
F9	Domain_registeration_length	F25	DNS Record
F10	Favicon	F26	Website Traffic
F11	Using Non-Standard Port	F27	PageRank
F12	HTTPS_token	F28	Google Index
F13	Request_URL	F29	Number of Links Pointing to Page
F14	URL_of_Anchor	F30	Statistical-Reports Based Feature
F15	Links_in_tags	F31	Result
F16	Server Form Handler (SFH)		

4898 phishing websites). The specific features are shown in Table 1. Although the UCI dataset is widely used, it is now too old to be used for modern phishing detection algorithms development.

B. MENDELEY_2018

48 features are contained in the dataset Mendeley_2018, which includes 5000 malicious and 5000 legitimate instances. The legal websites are derived from Alexa and common crawl, whereas phishing instances are from PhishTank and OpenPhish. Based on this dataset, L. Chiew et al. [19] proposed the HEFS framework mentioned in Section II. Table 2 shows a list of features in Mendeley_2018.

C. MENDELEY_2020

Dataset Mendeley_2020 is the primary dataset utilized in our research, which consists of two sub-datasets: dataset_full and dataset_small. There are 88647 instances in the full dataset and 58645 instances in the small dataset. Data were collected from PhishTank and Alexa ranking. This dataset contains 111 features, for better understanding, we redivided them into 8 groups. Two sub-datasets are illustrated in FIGURE 3, and the descriptions are explained in Table 3.

D. COMPARISON

Comparisons among the three datasets are provided in Table 4 and FIGURE 4. As shown in TABLE 4, there are more instances in dataset Mendeley_2020, even eight times as many as in datasets UCI_2015 and Mendeley_2018. In addition, all features in dataset UCI_2015 were transformed into Boolean type based on specified rules, making it difficult for further analysis. Dataset Mendeley_2020 was selected in our research for its quantity in instances and features.

IV. ML-BASED PHISHING DETECTION RESULTS

In this section, we performed an empirical analysis of various traditional ML algorithms for phishing detection.

TABLE 2. Features in dataset Mendeley_2018.

F1	NumDots	F25	NumSensitiveWords
F2	SubdomainLevel	F26	EmbeddedBrandName
F3	PathLevel	F27	PctExtHyperlinks
F4	UrlLength	F28	PctExtResourceUrls
F5	NumDash	F29	ExtFavicon
F6	NumDashInHostname	F30	InsecureForms
F7	AtSymbol	F31	RelativeFormAction
F8	TildeSymbol	F32	ExtFormAction
F9	NumUnderscore	F33	AbnormalFormAction
F10	NumPercent	F34	PctNullSelfRedirectHyperlinks
F11	NumQueryComponents	F35	FrequentDomainNameMismatch
F12	NumAmpersand	F36	FakeLinkInStatusBar
F13	NumHash	F37	RightClickDisabled
F14	NumNumericChars	F38	PopUpWindow
F15	NoHttps	F39	SubmitInfoToEmail
F16	RandomString	F40	IframeOrFrame
F17	IpAddress	F41	MissingTitle
F18	DomainInSubdomains	F42	ImagesOnlyInForm
F19	DomainInPaths	F43	SubdomainLevelRT
F20	HttpsInHostname	F44	UrlLengthRT
F21	HostnameLength	F45	PctExtResourceUrlsRT
F22	PathLength	F46	AbnormalExtFormActionR
F23	QueryLength	F47	ExtMetaScriptLinkRT
F24	DoubleSlashInPath	F48	PctExtNullSelfRedirectHyperlinks

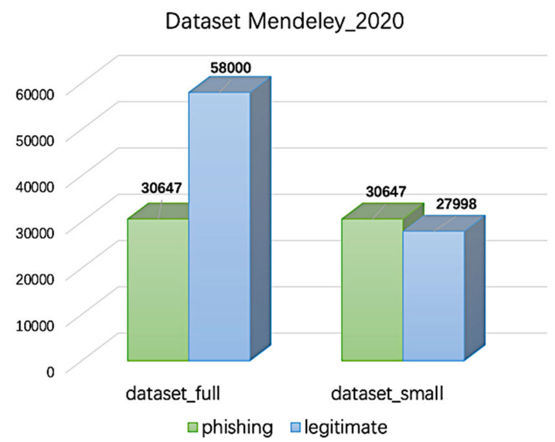

FIGURE 3. Dataset Mendeley_2020.

First, traditional ML algorithms including K-Means Clustering (KMeans), Support Vector Machine (SVM), Naive Bayes Classifier (NB), K-Nearest Neighbor (KNN), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART), and Random Forest (RF) were utilized to classify. Then, results by using ensemble ML methods including RF, AdaBoost, GBDT, XGBoost, and LightGBM were compared in the second sub-section. The same as most studies [4], [14] performance was analyzed using Accuracy, Precision, Recall, F1 score, ROC Curve, and P-R Curve.

TABLE 3. Features in dataset Mendeley_2020.

Group	No.	Description	Type
1	1-17	each number of“.-_/?=@&!~,+*#”$%” signs in the whole URL	Numeric
2	18-34	each number of“.-_/?=@&!~,+*#”$%” in domain	Numeric
3	35-51	each number of“.-_/?=@&!~,+*#”$%” in directory	Numeric
4	52-68	each number of“.-_/?=@&!~,+*#”$%” in file	Numeric
5	69-85	each number of“.-_/?=@&!~,+*#”$%” in parameters	Numeric
6	86-96	number of vowels, number of parameters, time_response, asn_ip, time_domain_activation, time_domain_expiration, number of resolved Ips, number of resolved NS, number of MX servers, Time-To-Live, number of redirects	Numeric
7	97-102	Top-level domain character length, number of characters in the whole URL, number of domain characters, number of directory characters, number of file characters, number of parameters characters	Numeric
8	103-111	is email present, is URL domain in IP address format, is “server” or “client” in domain, is TLD present in parameters, is domain has SPF, is URL has valid TLD/SSL certificate, is URL indexed on Google, is domain indexed on Google, is URL shortened	Boolean

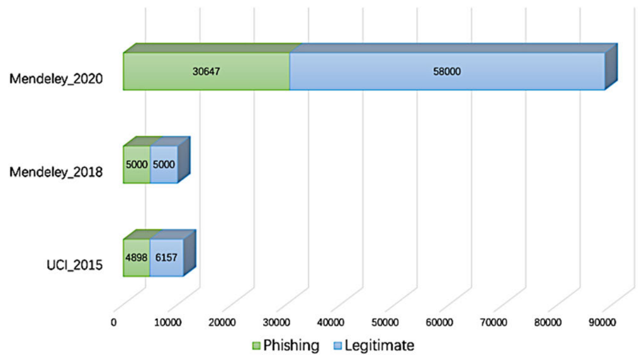

FIGURE 4. Number of instances in three phishing datasets.

A. TRADITIONAL ML ALGORITHMS

On Jupyter Notebook (6.4.3), all of the models were trained using the scikit-learn (1.1.2) library with Python (3.8.11) programming language. We used 10-fold cross-validation in our studies on the full dataset in Mendeley_2020. The performances are provided in Table 5, ROC Curves and P-R Curves are illustrated in FIGURE 5 and FIGURE 6. As a result, RF shows the best performance on all metrics with a 97.01% accuracy rate. As can be seen from the graphs, the highest value of Area Under Curve (AUC) belongs to RF, which means that it can separate the positive class and negative class correctly. Besides, RF presents the ability to

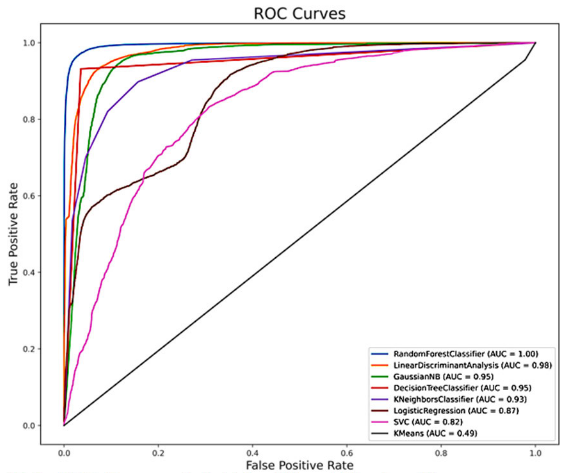

FIGURE 5. ROC curves of eight traditional ML classifiers.

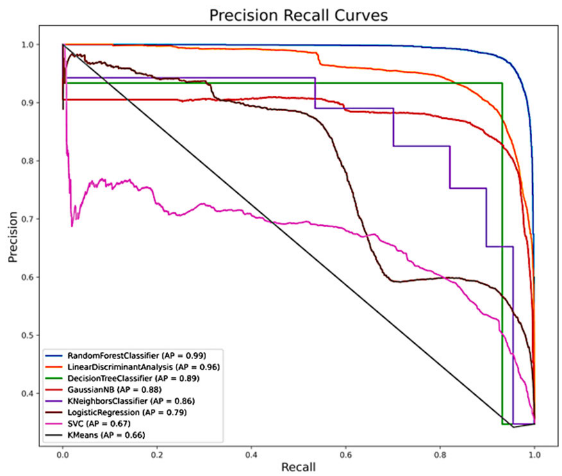

FIGURE 6. P-R curves of eight traditional ML classifiers.

return accurate results (high precision), as well as high positive results (high recall) at the same time in P-R Curves.

B. ENSEMBLE ML ALGORITHMS

The learning algorithms known as ‘‘ensemble ML methods’’ classify new data by performing a (weighted) vote on the predictions made by each classifier [39]. They are considered as the state-of-the-art solutions for many ML challenges [40]. We implemented 5 ensemble ML methods on the dataset including AdaBoost, Gradient Boosted Decision Trees (GBDT), LightGBM (version 3.3.3), Histogram-Based Gradient Boosting (HGB), and the most popular ensemble method Random Forest (RF). In this experiment, we split the original dataset into two parts, using 70% for training and 30% for testing.

Performances are provided in Table 6 and ROC curves are illustrated in FIGURE 7, where RF outperforms other methods in both accuracy rate and AUC value. LightGBM shows its high efficiency with minimum training and testing time consumption. We can conclude that ensemble ML methods,

TABLE 4. Comparison of three popular phishing datasets.

Dataset	Number of instances	Legitimate websites	Phishing websites	Number of features	Type of features	Features extracted From URL	Extra features
UCI_2015	11055	6157	4898	30	Boolean	12	18
Mendeley_2018	10000	5000	5000	48	Hybrid	25	23
Mendeley_2020	88647	58000	30647	111	Hybrid	96	14

TABLE 5. Performance metrics of various traditional ML algorithms.

No	Classifier	Accuracy(%)	Precision(%)	Recall(%)	F1score(%)
1	KMeans	62.60	51.67	13.78	16.96
2	SVM	75.46	67.30	55.89	61.06
3	NB	83.85	87.98	61.48	72.37
4	KNN	86.95	81.72	80.00	80.85
5	LR	89.76	87.38	82.27	84.59
6	LDA	91.54	82.77	95.26	88.58
7	CART	95.16	93.01	92.88	92.98
8	RF	97.01	95.44	95.93	95.69

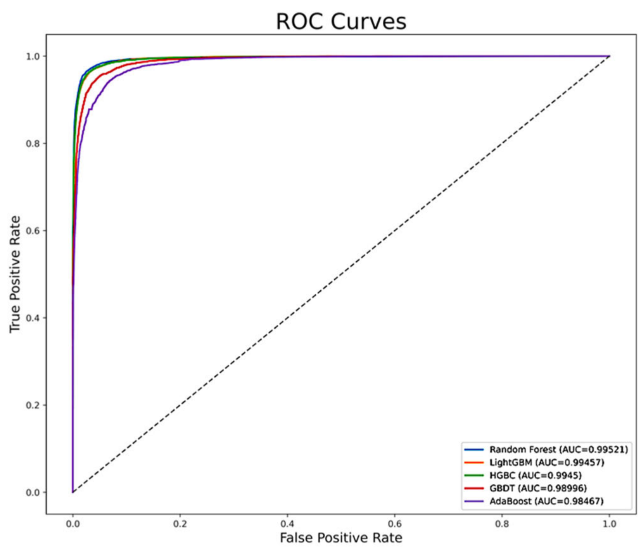

FIGURE 7. ROC curves of five ensemble ML classifiers.

in particular the boosting methods, tend to achieve the best performance in phishing classification.

V. DL-BASED PHISHING DETECTION RESULTS

The goal of this section is to assess the performance of current popular DL-based methods including FCNN, LSTM, and CNN. Fully Connected Neural Networks (FCNN) are constituted by a sequence of completely connected layers that have the primary advantage of being "structure agnostic," meaning that no special assumptions about the input are required [41]. LSTM is a particularly unique type of Recurrent Neural Network (RNN) that performs significantly better than the normal version. It was introduced by Hochreiter and Schmidhuber [42] and several researchers have since improved and popularized it. LSTMs are specifically designed to prevent the long-term dependency problem [43]. CNN is renowned for its ability to recognize simple patterns in a multi-dimensional task, and as a result, it has had success processing 2D signals like images and video frames [25]. However, a 1D CNN model can also be used to process datasets with a one-dimensional structure. [44]. In the following subsections, the experiment setup and data division are described, following the result and comparison.

A. EXPERIMENTAL SETUP

We built three DL-based models by using Python (3.8.11) with Tensorflow (2.9.1) and Keras library (2.9.0) on Jupyter Notebook (6.4.3). The dataset was divided into three parts: training dataset, validation dataset, and test dataset. The train dataset is 80% of the original dataset, and 20% is the test dataset. Furthermore, 10% of the train dataset is used as a validation dataset shown in FIGURE 8.

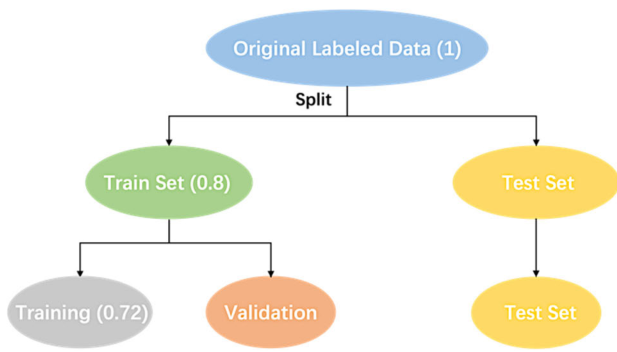

FIGURE 8. Dataset is divided into three parts.

Fully connected layers are usually used for classification, in order to build the FCNN model, it is essential to decide the number of layers, we set different layers to observe the changes in accuracy and loss on the validation dataset as shown in FIGURE 9. When the number of layers rises, the accuracy rate and loss are basically flat, and the validation accuracy rate is at its highst (0.9403) when the number of layers is 3.

Overfitting occurs when the number of layers is 20 in FIGURE 10, which indicates that the model fits perfectly against its training data but fails to perform accurately against the unseen (test) dataset, violating its purpose.

We built our 3-layers FCNN model after determining the epochs by using early stopping (FIGURE 11). The final model could be illustrated in FIGURE 12.

TABLE 6. Performance metrics of various ensemble ML algorithms.

No	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Training time cost (s)	Testing time cost (s)
1	AdaBoost	93.53	90.73	90.51	90.62	7.373	0.292
2	GBDT	95.33	92.95	93.57	93.26	32.128	0.074
3	HGB	96.54	94.93	95.17	95.17	3.491	0.078
4	LightGBM	96.60	94.90	95.27	95.09	0.742	0.054
5	RF	96.94	95.24	95.83	95.49	7.229	0.462

TABLE 7. Parameters settings for the three DL-Based models.

Model	Layers	Batch size	No of epochs	Optimizer	Activation function in hidden layers	Activation function in output layer
FCNN	3	32	22			
LSTM	3	32	32	adam	relu	sigmoid
CNN	6	32	16			

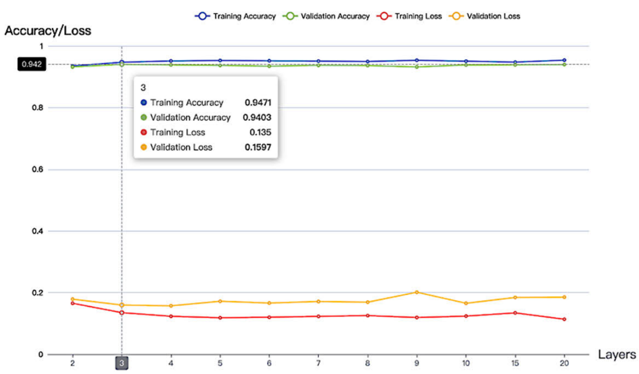

FIGURE 9. Accuracy and loss vs. number of layers in FCNN.

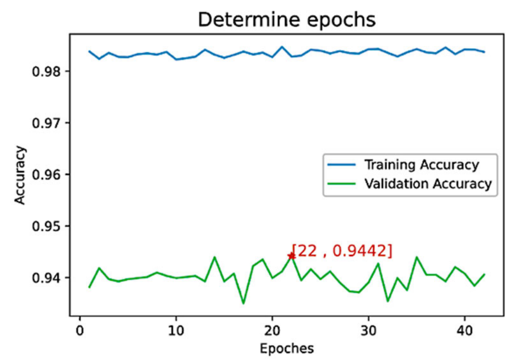

FIGURE 11. Accuracy vs. epochs in the 3-layers FCNN model.

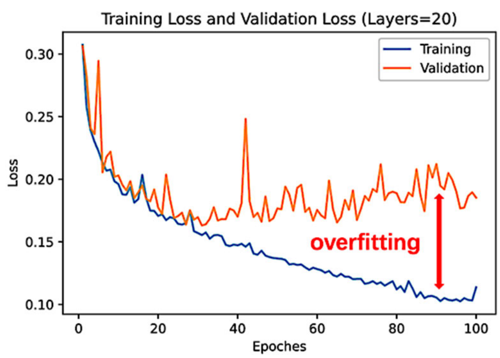

FIGURE 10. Overfitting occurs in the 20-layers FCNN model.

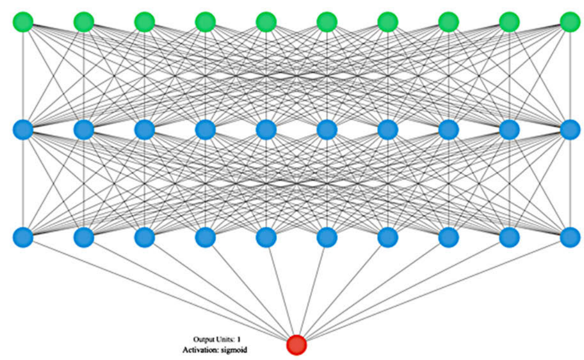

FIGURE 12. Our 3-layers FCNN model.

Procedure from FIGURE 9 to FIGURE 12 can be seen as a basic example of parameter settings in DL-based methods. Parameters can differ between different DL models, such as the number of layers in the model, batch size, the number of epochs, type of optimizer, type of activation function in hidden layers and output layer, etc. [4]. Based on these steps, we built a 3-layers LSTM model with one dropout layer and one dense layer. In addition, a 6-layers CNN model was constructed in the research. Table 7 lists the parameter settings for these DL architectures.

B. RESULT AND COMPARISON

To increase the reliability of classifications, models include RF were tested on three datasets: dataset_small with 111 features, dataset_full with 111 features, and dataset_full with 14 selected features in our previous work. For the purpose of seeing accuracy and loss during training process and validation process, accuracy and loss curves are illustrated in FIGURE13, where the upper graph shows accuracy and the lower graph shows loss function. As the number of epochs increases, the accuracy appears to rise but the loss

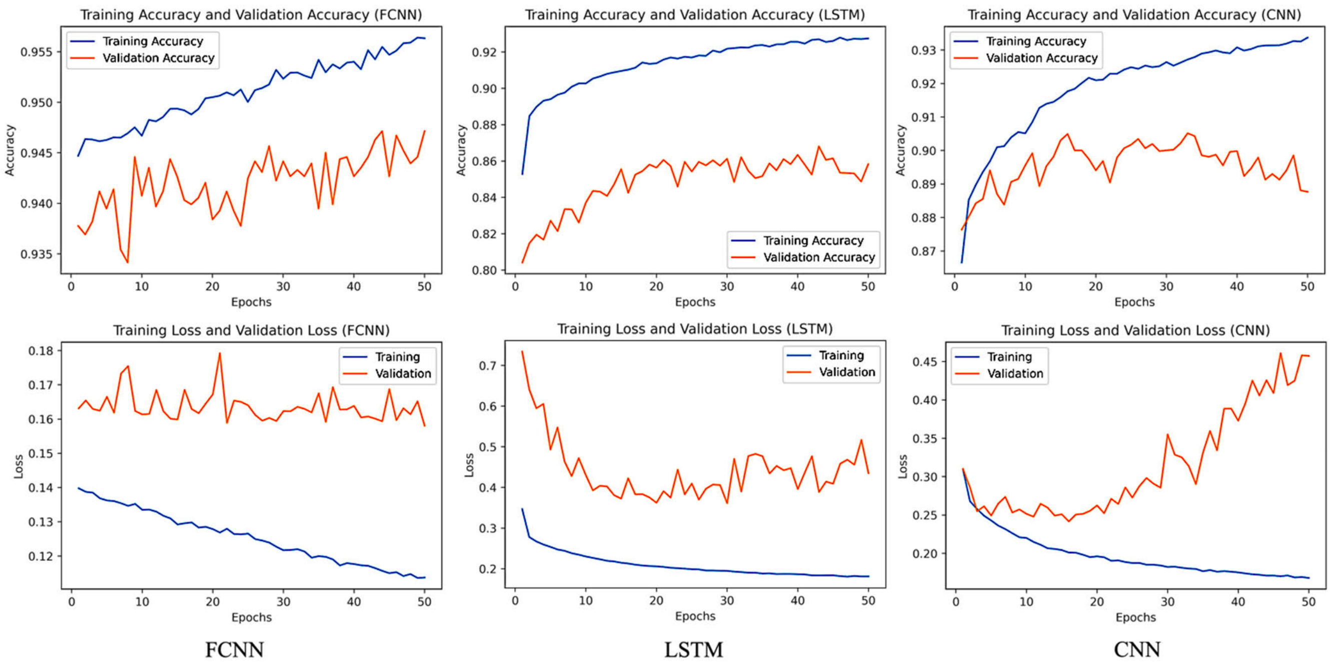

FIGURE 13. Accuracy and loss of FCNN, LSTM, and CNN.

TABLE 8. Performance metrics of RF, FCNN, LSTM, and CNN.

Dataset	No of instances	No of features	Model	Training time cost (s)	Precision (%)	Recall (%)	AUC (%)	Accuracy (%)
dataset_small	58645	111	RF	11.87	94.95	96.37	99.02	**95.41**
			CNN	306.82	91.00	90.46	96.84	**90.31**
			LSTM	140.72	81.18	97.69	96.81	**86.91**
			FCNN	74.77	81.12	95.09	95.23	**85.82**
dataset_full	88647		RF	15.43	95.55	95.55	99.50	**96.94**
			CNN	408.42	81.78	96.39	98.21	**91.38**
			LSTM	274.76	77.60	98.54	98.20	**89.73**
			FCNN	127.95	78.48	98.19	98.04	**90.13**
		14	RF	12.24	95.33	95.48	99.42	**96.84**
			CNN	116.05	78.07	90.43	95.33	**87.99**
			LSTM	289.18	68.18	98.83	97.18	**83.76**
			FCNN	95.09	65.86	98.62	96.36	**81.96**

function declines. A large gap between training outputs and validation outputs is commonly considered as overfitting, which typically happens when the model entirely memorizes data patterns, noise, and other random fluctuations, causing it fits too closely to the training set [45]. This phenomenon appears in CNN model visibly in FIGURE 13.

Table 8 summarizes the evaluation results acquired from the experiments. Evaluation metrics consist of training time consumption, precision, recall, AUC, and accuracy. From the table, we observed the following phenomenon that needs to be emphasized. First, all the classifiers perform better when data is getting bigger from dataset_small to dataset_full, which indicates that significant datasets are typically necessary for AI to reach high accuracy. Second, it is surprising that RF outperforms other DL models with the highest testing accuracy rate 96.94%, whereas that of CNN, FCNN, and LSTM are 91.38% 90.13%, and 89.73%, respectively. This result casts a new light on the performance of RF model. Third, RF model has the lowest training time, which is sensible because the computation complexity of DL-based models is always high. Note that we only record the training time cost of its best fine-tuning state for each individual model. Furthermore, we also conducted an experiment to compare the performances of the selected features against full features on dataset_full. Results showed that RF only experiences a minimal accuracy deterioration of 0.1% (96.94% to 96.84%) while achieving a massive reduction in the dataset. Compared to RF, DL models suffer from serious decreases in testing accuracy rate with selected features. FIGURE 14 also presents ROC Curves of the 4 classifiers, where lower plots are larger versions

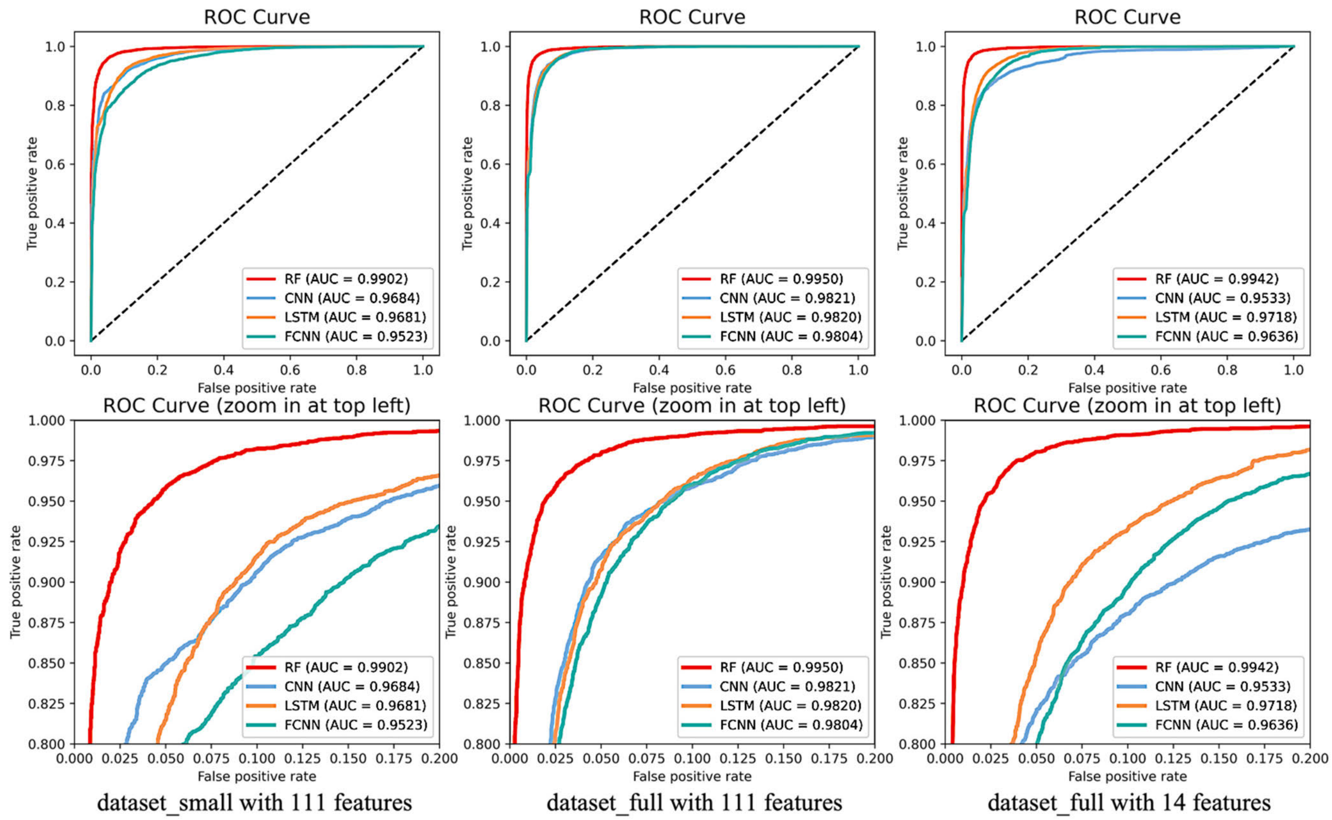

FIGURE 14. ROC curves of RF, CNN, LSTM, and FCNN on three different datasets.

zooming in at the top left. The curves and Area Under the ROC Curve (AUC) values offer a more comprehensive insight into the performances of the models. In every graph, RF clearly shows incomparable curves against other DL models.

As a result, the evaluation results have validated that RF is advantageous and highly effective when working with selected features and real-time applications in distinguishing between legitimate and phishing websites. The implications of these findings are discussed in the following Section to highlight the sufficiency of ensemble ML methods in phishing detection and navigate the future directions.

VI. DISCUSSION AND CONCLUSION

Previous sections have compared classification performances of various ML models and DL models. In this Section, we discuss the advantages and disadvantages between the two groups and draw our conclusion.

Deep Learning is considered to be the state-of-the-art solution to various problems with the advantages of dealing with big data and generating features automatically over Machine Learning. However, model architecture design, manual parameter tuning, high training time costs, computational complexity, and deficient accuracy performance are the most prevalent problems with DL approaches, as discussed in Section V.

Ensemble ML techniques represented by RF are usually regarded as a crystallization of wisdom of various ML methods. In ensemble methods, by combining different models, the risk of selecting an improper decision is reduced, and thus, the forecast performance is improved. In our experiments, CART, RF, and Boosting methods obtain better performances in phishing classification. This is potentially due to these ensemble methods benefit from the dynamic changing of assigned weight to each instance in the iteration process, making it more robust and stable than traditional ML algorithms. For instance, AdaBoost's basic principle is to concentrate on cases that were previously incorrectly classified when training a new inducer [40]. In the initial iteration, each instance is given the same weight, after which the weights of incorrectly categorized instances increase and those of correctly identified examples decrease. Additionally, based on their total prediction performances, the individual basic learners are also given voting weights. Hence, ensemble ML methods decrease both bias and variance of variable techniques while increasing the variance for stable classifiers, making them more suitable for classification tasks.

As a typical binary classification problem, ML-based phishing detection solutions are questioned on the ability to handle big data and extract features. Researchers believe that the process of feature selection relies on professional knowledge and reduplicative experiments, which is considered to be tedious, labor-intensive, and susceptible to human mistakes [4]. However, this problem can be effectively and efficiently resolved by utilizing automatic feature selection methods, for example, our feature selection framework achieves a remarkable 87.6% reduction in feature quantity with suffering from only a 0.1% deterioration in detecting accuracy, making it possible for up-date training and real-time detecting in a production environment. In another hand, phishers are also employing the latest schemes to execute attacks, phishing features are under evolution constantly. The phishing websites features cannot be generated once and for all, conversely, it should be a continuous updating and accumulating process, in which researchers are supposed to pay efforts.

To sum up, our experiments and discussions offer a significant insight into the sufficiency of ensemble ML methods for anti-phishing techniques. As for future work, we will validate our conclusion on various datasets with more features and more instances. In addition, further efforts need to be taken to avoid the inefficiency when detecting zero-day attacks. We plan to extract features of the latest phishing websites and train our ensemble ML method at intervals. Then, by observing the variation trends in newly evolving phishing patterns, we would like to find a balanced renewal frequency for extracting features and training models to maintain high detection accuracy. Last but not least, as a practical tool, a phishing detection architecture is supposed to be deployed in a real-world production environment (e.g. web browser) to verify its effectiveness against phishing attacks eventually.

REFERENCES

[1] N. Akdemir and S. Yenal, "How phishers exploit the coronavirus pandemic: A content analysis of COVID-19 themed phishing emails," *SAGE Open*, vol. 11, no. 3, Jul. 2021, Art. no. 21582440211031880, doi: 10.1177/21582440211031879.

[2] A. F. Al-Qahtani and S. Cresci, "The COVID-19 scamdemic: A survey of phishing attacks and their countermeasures during COVID-19," *IET Inf. Secur.*, vol. 16, no. 5, pp. 324–345, Sep. 2022, doi: 10.1049/ise2.12073.

[3] *APWG | Phishing Activity Trends Reports*. Accessed: Sep. 28, 2022. [Online]. Available: https://apwg.org/trendsreports/

[4] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[5] *Phishing Websites Features.pdf*. Accessed: Sep. 28, 2022. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf

[6] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, pp. 1–11, May 2016, doi: 10.1186/s13635-016-0034-3.

[7] A. A. Zuraiq and M. Alkasassbeh, "Review: Phishing detection approaches," in *Proc. 2nd Int. Conf. new Trends Comput. Sci. (ICTCS)*, Oct. 2019, pp. 1–6, doi: 10.1109/ICTCS.2019.8923069.

[8] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: Zero-day phishing website detection by visual similarity," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2020, pp. 1681–1698, doi: 10.1145/3372297.3417233.

[9] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Comput. Appl.*, vol. 33, pp. 1–17, Aug. 2021, doi: 10.1007/s00521-021-06401-z.

[10] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," 2020, *arXiv:2009.11116*.

[11] H. Tupsamudre, A. K. Singh, and S. Lodha, "Everything is in the name—A URL based approach for phishing detection," in *Cyber Security Cryptography and Machine Learning* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2019, pp. 231–248, doi: 10.1007/978-3-030-20951-3_21.

[12] E. S. Aung and H. Yamana, "URL-based phishing detection using the entropy of non-alphanumeric characters," in *Proc. 21st Int. Conf. Inf. Integr. Web-based Appl. Services*, New York, NY, USA, Dec. 2019, pp. 385–392, doi: 10.1145/3366030.3366064.

[13] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, no. 6, p. 154, Jun. 2021, doi: 10.3390/fi13060154.

[14] 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.

[15] U. Ozker and O. K. Sahingoz, "Content based phishing detection with machine learning," in *Proc. Int. Conf. Electr. Eng. (ICEE)*, Sep. 2020, pp. 1–6, doi: 10.1109/ICEE49691.2020.9249892.

[16] A. K. Jain, S. Parashar, P. Katare, and I. Sharma, "PhishSKaPe: A content based approach to escape phishing attacks," *Proc. Comput. Sci.*, vol. 171, pp. 1102–1109, Jan. 2020, doi: 10.1016/j.procs.2020.04.118.

[17] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, "An adaptive machine learning based approach for phishing detection using hybrid features," in *Proc. 5th Int. Conf. Web Res. (ICWR)*, Apr. 2019, pp. 281–286, doi: 10.1109/ICWR.2019.8765265.

[18] R. Zaimi, M. Hafidi, and M. Lamia, "Survey paper: Taxonomy of website anti-phishing solutions," in *Proc. 7th Int. Conf. Social Netw. Anal., Manag. Secur. (SNAMS)*, Dec. 2020, pp. 1–8, doi: 10.1109/SNAMS52053.2020.9336559.

[19] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci.*, vol. 484, pp. 153–166, May 2019, doi: 10.1016/j.ins.2019.01.064.

[20] Y. Wei and Y. Sekiya, "Feature selection approach for phishing detection based on machine learning," in *Proc. Int. Conf. Appl. CyberSecurity (ACS)*, 2021, pp. 61–70, doi: 10.1007/978-3-030-95918-0_7.

[21] S. Al-Ahmadi. (2020). *PDMLP: Phishing Detection Using Multilayer Perceptron*. Rochester, NY, USA. Accessed: Sep. 1, 2022. [Online]. Available: https://papers.ssrn.com/abstract=3624621

[22] A. Odeh, I. Keshta, and E. Abdelfattah. (2020). *Efficient Detection of Phishing Websites Using Multilayer Perceptron*. International Association of Online Engineering. Accessed: Sep. 30, 2022. [Online]. Available: https://www.learntechlib.org/p/217754/

[23] Q. Li, M. Cheng, J. Wang, and B. Sun, "LSTM based phishing detection for big email data," *IEEE Trans. Big Data*, vol. 8, no. 1, pp. 278–288, Feb. 2022, doi: 10.1109/TBDATA.2020.2978915.

[24] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Deep learning with convolutional neural network and long short-term memory for phishing detection," in *Proc. 13th Int. Conf. Softw., Knowl., Inf. Manag. Appl. (SKIMA)*, Aug. 2019, pp. 1–8, doi: 10.1109/SKIMA47702.2019.8982427.

[25] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Mar. 2020, pp. 1–6, doi: 10.1109/ICCAIS48893.2020.9096869.

[26] Y. Su, "Research on website phishing detection based on LSTM RNN," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Jun. 2020, pp. 284–288, doi: 10.1109/ITNEC48623.2020.9084799.

[27] T. Feng and C. Yue, "Visualizing and interpreting RNN models in URL-based phishing detection," in *Proc. 25th ACM Symp. Access Control Models Technol.*, Jun. 2020, pp. 13–24, doi: 10.1145/3381991.3395602.

[28] Y. Lin. (2021). *Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages*. Accessed: Sep. 30, 2022. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/lin

[29] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Wozniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Netw.*, vol. 178, Sep. 2020, Art. no. 107275, doi: 10.1016/j.comnet.2020.107275.

[30] S. Mahdavifar and A. A. Ghorbani, "DeNNeS: Deep embedded neural network expert system for detecting cyber attacks," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14753–14780, 2020, doi: 10.1007/s00521-020-04830-w.

[31] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, Jun. 2019, doi: 10.1016/j.neucom.2019.02.056.

[32] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, "SoK: A comprehensive reexamination of phishing research from the security perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 671–708, 1rst Quart., 2020, doi: 10.1109/COMST.2019.2957750.

[33] *UCI Machine Learning Repository: Phishing Websites Data Set*. Accessed: Oct. 1, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[34] C. L. Tan, "Phishing dataset for machine learning: Feature evaluation," *Mendeley Data*, vol. 1, Mar. 2018, doi: 10.17632/h3cgnj8hft.1.

[35] G. Vrbancic, I. Fister, and V. Podgorelec, "Datasets for phishing websites detection," *Data Brief*, vol. 33, Dec. 2020, Art. no. 106438, doi: 10.1016/j.dib.2020.106438.

[36] *PhishTank | Join the Fight Against Phishing*. Accessed: Oct. 1, 2022. [Online]. Available: https://phishtank.org/

[37] G. H. Lokesh and G. BoreGowda, "Phishing website detection based on effective machine learning approach," *J. Cyber Secur. Technol.*, vol. 5, no. 1, pp. 1–14, Jan. 2021, doi: 10.1080/23742917.2020.1813396.

[38] A. Lakshmanarao, P. S. P. Rao, and M. M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 1164–1169, doi: 10.1109/ICAIS50930.2021.9395810.

[39] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Germany: Springer, 2000, pp. 1–15, doi: 10.1007/3-540-45014-9_1.

[40] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, Jul. 2018, doi: 10.1002/widm.1249.

[41] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4580–4584, doi: 10.1109/ICASSP.2015.7178838.

[42] *Long Short-Term Memory | Neural Computation | MIT Press*. Accessed: Oct. 2, 2022. [Online]. Available: https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory

[43] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306, doi: 10.1016/j.physd.2019.132306.

[44] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107398, doi: 10.1016/j.ymssp.2020.107398.

[45] X. Ying, "An overview of overfitting and its solutions," *J. Phys., Conf.*, vol. 1168, Feb. 2019, Art. no. 022022, doi: 10.1088/1742-6596/1168/2/022022.

YI WEI received the B.E. degree from the College of Computer Science and Electronic Engineering, Hunan University, China, in 2018, and the M.E. degree in electrical engineering and information systems from The University of Tokyo, Tokyo, Japan, in 2021, where she is currently pursuing the Ph.D. degree in electrical engineering and information systems.

Since August 2021, she has been a Technical Assistant with the Security Informatics Education and Research Center and the Graduate School of Information Science and Technology, The University of Tokyo. Her research interests include phishing website detection by using machine learning and deep learning, feature selection approach for dimensionality reduction, and applications of future quantum machine learning algorithms in the cybersecurity field.

YUJI SEKIYA received the B.E. degree from Kyoto University, in 1997, and the M.E. degree and the Ph.D. degree in media and governance from Keio University, Tokyo, Japan, in 1999 and 2005, respectively.

Since October 1999, he has been working as a Visiting Researcher at USC/ISI for six months. Since 2002, he has also been working at the Information Technology Center, The University of Tokyo, where he is currently a Professor at the Graduate School of Information Science and Technology and working as a member of the Security Informatics Education and Research Center. He has been working on DNS measurements and security, SDN, network virtualization, cloud computing, and cyber security. As society activities, he is deeply involved in WIDE Project, M Root DNS server, JP DNS servers, Internet Exchanges called, DIX-IE, PIX-IE, and NSPIXP-3, NECOMA Project, and Interop Tokyo ShowNet. He has been in-charge of the Executive Advisor to the Japanese Government CIO, since February 2020, and also in-charge of a Senior Network Engineer at Digital Agency of Japanese Government.