IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 20, 2022, accepted March 17, 2022, date of publication March 28, 2022, date of current version April 4, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3162588*

# Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing

**HATMA SURYOTRISONGKO[1,2], (Member, IEEE), YASUO MUSASHI[3], (Member, IEEE), AKIO TSUNEDA[4], (Member, IEEE), AND KENICHI SUGITANI[3]**

[1]Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan
[2]Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[3]Center for Management of Information Technologies, Kumamoto University, Kumamoto 860-8555, Japan
[4]Faculty of Advanced Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan

Corresponding author: Hatma Suryotrisongko (hatma@is.its.ac.id)

This work was supported by the Kumamoto University's Center for Management of Information Technology (CMIT) researchers and staff members, and professors in the Faculty of Advanced Science and Technology.

**ABSTRACT** We investigated 12 years DNS query logs of our campus network and identified phenomena of malicious botnet domain generation algorithm (DGA) traffic. DGA-based botnets are difficult to detect using cyber threat intelligence (CTI) systems based on blocklists. Artificial intelligence (AI)/machine learning (ML)-based CTI systems are required. This study (1) proposed a model to detect DGA-based traffic based on statistical features with datasets comprising 55 DGA families, (2) discussed how CTI can be expanded with computable CTI paradigm, and (3) described how to improve the explainability of the model outputs by blending explainable AI (XAI) and open-source intelligence (OSINT) for trust problems, an antidote for skepticism to the shared models and preventing automation bias. We define the XAI-OSINT blending as aggregations of OSINT for AI/ML model outcome validation. Experimental results show the effectiveness of our models (96.3% accuracy). Our random forest model provides better robustness against three state-of-the-art DGA adversarial attacks (CharBot, DeepDGA, MaskDGA) compared with character-based deep learning models (Endgame, CMU, NYU, MIT). We demonstrate the sharing mechanism and confirm that the XAI-OSINT blending improves trust for CTI sharing as evidence to validate our proposed computable CTI paradigm to assist security analysts in security operations centers using an automated, explainable OSINT approach (for second opinion). Therefore, the computable CTI reduces manual intervention in critical cybersecurity decision-making.

**INDEX TERMS** Adversarial machine learning, botnet, cybersecurity, DGA, explainable artificial intelligence, threat intelligence.

## I. INTRODUCTION

Cyber threat intelligence (CTI), often also referred to as threat intelligence (TI), can be understood as processes, tools, and activities in the cyber world's intelligence cycle [1]. CTI also includes analytical processes used to analyze threats and share information regarding threats, such as indicators of compromise (IoC). Therefore, it is a crucial concept of future cybersecurity to recognize a large-scale cyberattack incident rapidly [1].

Since the mid-1980s, researchers have been highlighting the explainability aspects of artificial intelligence (AI) or expert systems, which grows into explainable AI (XAI) research field [2]–[4].

### A. BOTNET DGA DETECTION

Botnet is notorious for causing many severe cyberattacks. It uses a command and control (C&C) server [5]. Domain generation algorithm (DGA)-based botnet is arguably the most challenging type of botnets to detect because it uses various C&C server domains that are algorithmically generated. The bot queries a C&C server through a domain name system (DNS) traffic to hide their communication messages. Therefore, security defense systems find it challenging to detect [6].

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

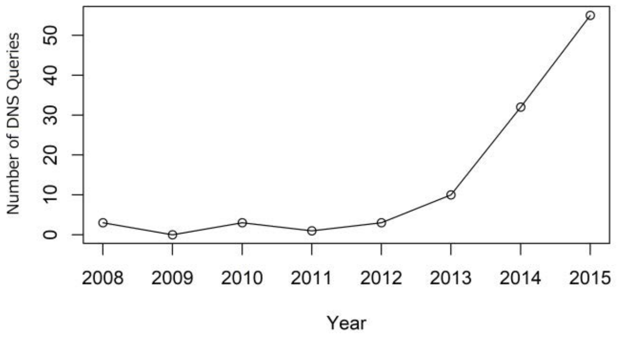


**FIGURE 1.** Changes in the malicious botnet DGA traffics (Conficker) in our campus networks, observed by DNS query logs analysis.

We analyzed a 12-year dataset of DNS server query logs (4,383 days from 2004 to 2015) from our campus' full cache resolver. Surprisingly, 107 DNS queries (malicious domain names) were detected as botnet traffic (Fig. 1). These results were obtained using 803,333 malicious domain names from our blocklist dataset; however, due to the characteristics of the algorithmically generated domain in the DGA [6], depending only on a domain blocklist might be insufficient. These characteristics indicate that an AI/machine learning (ML) model for CTI systems is necessary.

In this paper, we developed a botnet DGA classification model and provided a comparison with a recent previous work [7] which used the same approach (statistical features-based) and same ML algorithm (random forest). We also compared our model's robustness with previous works of character-based deep learning models [8] against three state-of-the-art DGA adversarial ML attacks (MaskDGA [9], CharBot [10], and DeepDGA [11]). Furthermore, using the DGA detection problem as a case study, we demonstrated our approach to extend the current CTI sharing paradigm, described in the next section.

### B. CTI SHARING PARADIGM

Literature surveys/reviews of CTI sharing are available in [1], [12], [13]. To the best of our knowledge, no research highlighting the importance of sharing AI/ML models for CTI has been published. Moreover, existing CTI platforms [12] and tools [13] (e.g., MISP, NC4 CTX/Soltra Edge, ThreatConnect, AlienVault, IBM X-Force Exchange, Anomali, ThreatExchange, CrowdStrike, ThreatQuotient, EclecticIQ, CRITs, CIF v3, AlliaCERT) do not appear to promote sharing AI/ML models for CTI.

Therefore, this research aims to fill this gap by proposing an extension of actionable CTI, namely, computable CTI, a new paradigm in CTI sharing. We define computable CTI as the next level of actionable CTI by extending the European Union Agency for Cybersecurity (ENISA)'s definition of actionable CTI using AI/ML computability criteria [14]. Furthermore, we define that computable CTI paradigm encourages sharing AI or ML models of CTI systems for cybersecurity communities. As there are already marketplaces for AI models, how to achieve the concept of extending CTI sharing with AI/ML models in a practical manner will be meaningful. The challenging issues with broad adoption of computable CTI sharing include potential bias of decision, privacy preservation [15], [16] and robustness against adversarial ML attacks [17], [18].

Fig. 2 shows the conceptual design of computable CTI. Various CTI sources are available in the market and even publicly available to communities, such as open-source intelligence (OSINT) [19]. Our proposed paradigm uses the OSINT ecosystem to enhance XAI techniques by providing a second opinion from IoC obtained from OSINT, thus preventing automation bias when using AI/ML for security automation of CTI applications. The interaction works two ways: retrieving IoC from OSINT for a second opinion and submitting new confirmed IoC findings to OSINT repositories. The submission of new information to OSINT repositories needs to be conducted in a careful manner, as it can be a way to inject false or poor quality findings intentionally or unintentionally that can cause issues with other detectors relying on such OSINT and poison any training process [20].

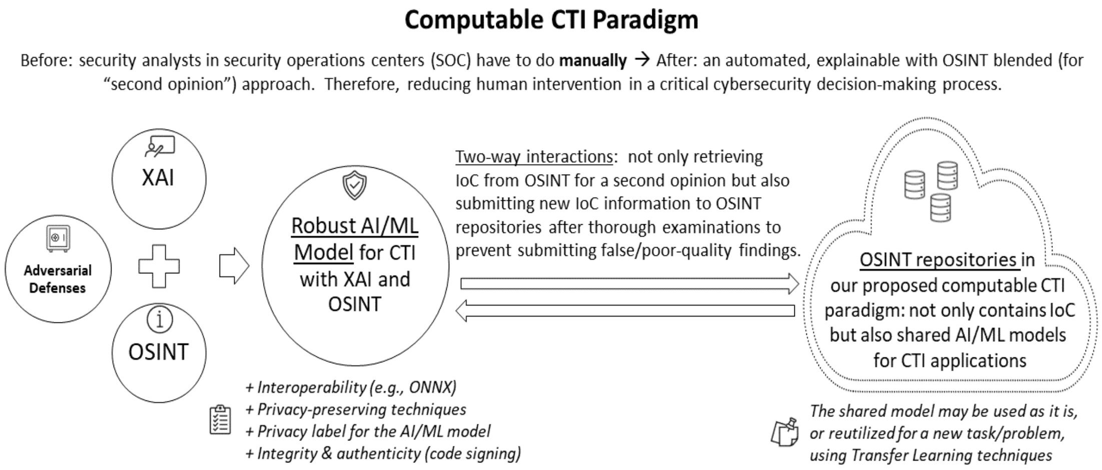
As highlighted in [21] and [22], trust is a critical ingredient in the CTI sharing ecosystem. The increasing popularity of OSINT, where communities can subscribe and add new IoC of malicious malware, dangerous domain names/IP addresses, and other information related to a threat, increases the concern of trust and validity in CTI because fake/false IoC information can be quickly submitted into OSINT repositories [19].

Recently, XAI has become an important copilot assisting human users and experts in making critical decisions [23]. XAI could give leverage to make serious decisions in the medical domain [24] and security decisions in dealing with cyber threats in a complicated/mission-critical situation. Because explainability is being mandated in the European Union General Data Protection Regulation (GDPR), it has become critical for practitioners across industries [25]. However, as stressed in [26], explanations in the XAI implementation should be tailored depending on the context and other considerations. Achieving trustable XAI is still one of the grand challenges being pursued by researchers in this field [27], [28]. Our research's objective is to propose blending XAI and OSINT to solve this problem of trust.

### C. CONTRIBUTIONS

First, this study expands CTI with computable CTI to reduce human intervention in the cybersecurity decision-making process [23], [24]. Moreover, we improve the explainability of the AI/ML model outputs by blending the XAI and OSINT methods to enhance trust in CTI sharing. We use the DGA detection problem to demonstrate a proof of concept and validate our proposed computable CTI paradigm.

Second, this research proposes a model to detect botnet DGA-based traffic. Compared with a recent study [7], which uses a similar approach (statistical features-based with random forest algorithm) but depends on 24 features,



Computable CTI Paradigm

Before: security analysts in security operations centers (SOC) have to do **manually** → After: an automated, explainable with OSINT blended (for "second opinion") approach. Therefore, reducing human intervention in a critical cybersecurity decision-making process.

Two-way interactions: not only retrieving IoC from OSINT for a second opinion but also submitting new IoC information to OSINT repositories after thorough examinations to prevent submitting false/poor-quality findings.

OSINT repositories in our proposed computable CTI paradigm: not only contains IoC but also shared AI/ML models for CTI applications

+ Interoperability (e.g., ONNX)
+ Privacy-preserving techniques
+ Privacy label for the AI/ML model
+ Integrity & authenticity (code signing)

The shared model may be used as it is, or reutilized for a new task/problem, using Transfer Learning techniques

**FIGURE 2. Conceptual design of the proposed computable CTI paradigm.**

using only seven features proposed in our paper is enough to deliver a satisfactory performance. Moreover, compared with character-based deep learning models [8], our proposed model provides a better defense against three state-of-the-art DGA adversarial ML attacks (CharBot [10], DeepDGA [11], MaskDGA [9]).

Third, this study contributes to the cybersecurity literature as a gateway/direction for future research on AI/ML-based CTI sharing. We released the code and datasets of this study to the IEEE Code Ocean [29] and IEEE DataPort [30] to facilitate reproducible research.

## II. RELATED WORKS

Although multiple approaches can be used to guard DNS systems against botnet malware (hiding its communications with C&C server), the conventional DNS security approach, which is filtering based on blocklists, is problematic because no blocklist is completely reliable [31], [32]. The most recent survey/review papers on DNS-based DGA-botnet malicious domain name detection can be seen in [5], [33], [34].

In a recent publication on botnet DGA detection, Hoang and Vu [7] proposed an improved random forest-based model by calculating 24 statistical features, such as character n-gram frequency distributions of a domain name, entropy values, the first character is a digit number or not, and various other statistical calculations. Their experimental results using a dataset of 39 DGA families show an enhanced performance compared with their previous works [7].

However, one may argue that calculating 24 features increases the complexity in computation. Thus, we proposed to use only seven features: entropy, relative entropy Alexa, the minimum of relative entropy botnets, information radius, character length, a new feature generated using a decision tree algorithm, and a domain name's reputation score. In addition, we experiment with broader coverage of DGA datasets (55 DGA families in total).

Besides the statistical features approach, the character-based classification approach, which relies on character-level embeddings of a domain name, can also be used for DGA classification. Yu *et al.* [8] compiled character-level-based deep learning models with various architectures for DGA classification. To evaluate our work, we implemented four deep learning models: Endgame [35], CMU [36], NYU [37], and MIT [38].

The current state-of-the-art DGA attacks can be found in CharBot [10], DeepDGA [11], and MaskDGA [9], where the authors employed various sophisticated approaches, such as adversarial ML evasion attack/adversarial examples, to generate domain names for evading DGA classifiers. Sidi *et al.* [9] demonstrated that MaskDGA attack reduces the performance of a DGA classifier, evading detection system. Peck *et al.* [10] showed the effectiveness of the CharBot attack, reducing the detection rate of a classifier as low as 1.69%; even retraining the classifiers is not a viable defense strategy.

We applied those three DGA attacks (CharBot, DeepDGA, and MaskDGA) to check the performance of our model for botnet DGA detection applications in dealing with harsh adversarial attacks.

To grasp the understanding of current state-of-the-art XAI, literature reviews and surveys are available in [39]–[43]. Table 1 shows our XAI method selection position, adopting the XAI taxonomy classification systems. The details of our proposed second opinion approach using the XAI-OSINT blend will be elaborated in the subsequent section. In this study, blending XAI and OSINT (as the second opinion) for the CTI system delivers a practical implication to solve trust in the computable CTI ecosystem, i.e., either lack of trust or too much trust (automation bias). The relation between explanation and trust is important [44].

## III. METHODS

### A. DATASETS

Using an ML algorithm to detect malicious DNS traffic requires accurate ground-truth data for both model training

**TABLE 1.** XAI Methods to Improve Trust for CTI Sharing.

| Taxonomy | ANCH-OR | LIME | SH-AP | Counter-factual | XAI-OSINT |
|---|---|---|---|---|---|
| ***Explanation by simplification:*** | | | | | |
| Rule-based learner | √ | √ | - | - | - |
| Decision tree | - | - | - | - | - |
| ***Feature relevance explanation:*** | | | | | |
| Influence function | - | - | - | - | - |
| Sensitivity | - | - | - | - | - |
| Game Theory inspired | - | - | √ | - | - |
| Interaction based | - | - | - | - | - |
| ***Local explanation:*** | | | | | |
| Rule-based learner | √ | √ | - | - | - |
| Decision tree | - | - | - | - | - |
| ***Visual explanation:*** | | | | | |
| Conditional/dependence/ Shapley plots | - | - | √ | - | - |
| Sensitivity/saliency | - | - | - | - | - |
| ***Explanation by example:*** | | | | | |
| Counterfactual explanation | - | - | - | √ | - |
| ***Our proposed approach:*** | | | | | |
| Second opinion | - | - | - | - | √ |

**TABLE 2.** Datasets With Total 55 Botnet DGA Families.

| DGA Family | Sample content of the dataset |
|---|---|
| abcbot | fuorhtpsx.tk |
| antavmu | 5f474370.com |
| bamital | a9d68c6203f04de3265bb8c7584e476b.info |
| banjori | igxbtallulahavaw.com |
| bigviktor | callleastrace.fans |
| blackhole | owrfrxrmiewneegp.ru |
| ccleaner | ab2ec634a79.com |
| chinad | 797nkllqgz9x7x28.com |
| conficker | dxtpafsyjcw.cc |
| copperstealer | dfd886470ec28240.com |
| cryptolocker | xeuskjcythfllwh.org |
| dircrypt | rvonetvypqacbsa.com |
| dyre | y243c1001a327885f71ee399229ef82609.cc |
| emotet | qijfcnekvhwvcgkg.eu |
| enviserv | 33aaf2199f.net |
| feodo | mgcdlsidwsdnolwzyz.ru |
| flubot | oupxbsfpvukowup.cn |
| fobber | ylbphjjdjs.com |
| gameover | 115vvgdobj3uf1jq8gi174q2t7.net |
| gspy | cc9cf6ae3922d07d.info |
| kfos | help-google.tw |
| locky | gcqbsfpxkhqf.tf |
| madmax | www.k3bdsbsa3k.net |
| matsnu | dishcow-catcondition.com |
| mirai | cmhewcvopvno.online |
| moneròminer | 5c95f79304b49.org |
| murofet | niqlkgqytoirnou.org |
| mydoom | hrsrrarpsr.net |
| necro | vlxdqiwafmbashxa.viewdns.net |
| necurs | hjvtlavpidi.su |
| ngioweb | subozirion-multirusenelike.com |
| nymaim | yowgbazj.pw |
| omexo | 8f86373028729e6497f00487d7775f81.net |
| padcrypt | efddmacncmndodcn.website |
| proslikefan | pbeuiykocu.ru |
| pykspa | zzzkdgn.org |
| qadars | 7g9ijc9e3why.net |
| qakbot | pnhwjybkdixb.com |
| ramnit | nnnfyqtv.com |
| ranbyus | gwoukqdssttrhg.me |
| rovnix | o4cwfxophfp7iiq4zs.biz |
| shifu | gemmpbt.eu |
| shiotob | sifulj2yl1g.com |
| simda | pufybyg.info |
| suppobox | coriandertimothyson.net |
| symmi | weaxabudodine.ddns.net |
| tempedreve | ozrnlglgb.info |
| tinba | wddyvorijopl.ru |
| tinynuke | dedc87d2095576f2842c7be426613667.com |
| tofsee | eaieaih.biz |
| tordwm | dab53527.top |
| vawtrak | mxubexcvqvc.com |
| vidro | wwcdjkdsg.dyndns.org |
| virut | jawukx.com |
| xshellghost | texcrgluvmrgr.com |

and accuracy evaluations [45]. For the first experiment in our study, we used Alexa Top 1M (1,000,000 domain names) and 803,333 domain names of ten botnet DGA families used in [46]: Conficker, Cryptolocker, Goz, Matsnu, New_Goz, Pushdo, Ramdo, Rovnix, Tinba, Zeus. Then, we employed 998,503 domain names of 55 DGA families from Netlab 360 [47] (Table 2 ).

## B. FEATURES

We analyzed the datasets by calculating entropy using Shannon's function (1) as our model's first feature. As reported in our previous publication [48], entropy fluctuations can indicate an increasing number of unique random query keywords in DNS queries, frequently observed during dangerous situations, such as Kaminsky-like attacks. Then, we extended a statistical measurement [49] using relative entropy (RE) via Kullback–Leibler divergence (2).

$$H(X) = -\sum_{i \in X} P(i) \log_2 P(i) \tag{1}$$

$$DKL(P||Q) = \sum_{i} p_i \log(\frac{p_i}{q_i}) \tag{2}$$

where $Q$ is the baseline distribution calculated on legitimate data (Alexa top 1M domains or ten botnet domains datasets) and $P$ is the target distribution (i.e., the domain in the DNS query log to be verified).

Our model's second feature is RE-Alexa, which measures the distance (or similarity) between the domain in question to Alexa domain unigram distributions. The third feature is Min-RE-Botnets. Here, we calculated the RE value of a suspicious domain with each botnet dataset and considered the minimum value as the Min-RE-Botnets value.

Inspired by Sharifnya's work [50], the fourth feature in the proposed model is the information radius (IRad) value, calculated using the Jensen–Shannon divergence function (3). This function is a generalization of the Jensen–Shannon divergence that compares more than two probability distributions. The proposed model uses this function to calculate a target domain name's distance to the botnet datasets.

$$JSD\pi_1, \ldots, \pi_n(P_1, \ldots, P_n) = H(\sum_{i=1}^{n} \pi_i P_i) - \sum_{i=1}^{n} \pi_i H(P_i) \qquad (3)$$

The next feature is the domain name character length (CharLength). This feature is suitable for botnet DGA detection because several DGA algorithms in our dataset demonstrate similar character lengths, characterizing a unique property of these randomly generated domains.

Then, a new feature is generated using a decision tree algorithm (TreeNewFeature). Here, we combined entropy, RE-Alexa, Min-RE-Botnets, and CharLength features using decision trees and used them to train a predictive model. We constructed a decision tree using those features and used the prediction result as a new feature.

The last feature is the Alexa reputation score (ReputationAlexa). This approach was inspired by Zhao's work [51]. Here, we used the Alexa Top 1M domains to generate a weight matrix to calculate a domain reputation value. The procedure to generate the weight matrix begins by reading all 1M domains from Alexa, and then learning the vocabulary dictionary of n-gram 3 to 5 characters and returning the term-document matrix. Note that we used the base-10 logarithm function of the total n-gram matrix of all Alexa 1M domains, as shown in (4).

$$W_{N-gram}(i) = \log_{10}\left(\sum_{i=1}^{n} C_{N-gram}(i)\right) \qquad (4)$$

where $W$ is the weight matrix used to calculate the reputation score and $C_{N-gram}$ is the character n-gram frequencies. When calculating a target domain's reputation score, we first extract token counts from the target domain using a vocabulary n-gram character constructor. This calculation is the same as generating a document-term matrix using Alexa Top 1M.

### C. EXPERIMENTS

We conducted three experiments:

1) We used multiple supervised ML algorithms to compare results to select an algorithm with the best accuracy. Here, five algorithms (naive Bayes, logistic regression, extra tree, random forest, and ensemble learning) were computed using the Scikit-Learn [52].
2) Comparing our random forest model with the latest previous work [7].
3) To check the performance of our model for botnet DGA detection in dealing with harsh adversarial attacks, we conducted a robustness evaluation of our classifier against three state-of-the-art DGA attacks (CharBot [10], DeepDGA [11], MaskDGA [9]) and comparison with four deep learning models (Endgame [35], CMU [36], NYU [37], MIT 38]).

The evaluation metric is given in (5), where *TP*, *TN*, *FP*, *FN* stands for true positive, true negative, false positive, and false negative, respectively. Table 3 summarizes the variables/parameters used in the random forest model.

**TABLE 3. Summary of parameters used in the random forest model.**

| Parameters | Values |
|---|---|
| bootstrap | True |
| ccp_alpha | 0.0 |
| class_weight | None |
| criterion | gini |
| max_depth | None |
| max_features | auto |
| max_leaf_nodes | None |
| max_samples | None |
| min_impurity_decrease | 0.0 |
| min_impurity_split | None |
| min_samples_leaf | 1 |
| min_samples_split | 2 |
| min_weight_fraction_leaf | 0.0 |
| n_estimators | 1500 |
| n_jobs | 40 |
| oob_score | False |
| random_state | 1 |
| verbose | 1 |
| warm_start | False |

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

### D. BLENDING XAI AND OSINT

This research applied four existing XAI techniques (ANCHOR, local interpretable model-agnostic explanations (LIME), Shapley additive exPlanations (SHAP), and counterfactual explanation) and proposed our approach (XAI and OSINT blend) to produce a second opinion explanation (Table 1).

We take advantage of the SHAP method for presenting a global explanation delivering a Game Theory-inspired feature relevance explanation [53]. SHAP is based on the game theoretically optimal Shapley values. We focus on using the model-agnostic approach to enable more freedom to use any advanced algorithm for a classification model. Therefore, we considered implementing the KernelExplainer, kernel-based estimation approach for Shapley values inspired by local surrogate models for SHAP explanation. Moreover, we use SHAP's force plot to provide a local explanation.

The next XAI method implemented in this study is the LIME [54]. LIME trains local surrogate models to explain individual prediction/classification. It provides a local explanation, explaining individual classification results of a black-box model. Therefore, users will understand why the CTI system classifies a suspected domain name into a legit or botnet DGA domain name.

Next, we applied ANCHORS, the LIME's improvement, to predict how a model would behave with less effort and higher precision [55]. ANCHORS is a rule-based learner, explaining by simplification. We expect an explanation expressed as easy-to-understand IF-THEN rules from this method. This type of expression might be more convenient to explain the model's behavior: why did the CTI system decide a domain name as a botnet DGA domain, or why the CTI system classified this suspicious-looking domain name as a legit domain name? We used Alibi [56] to implement ANCHORS in our CTI system.

Next, we applied a counterfactual explanation, adding explainability using an example. We used the What-If Tool [57] to implement this functionality, thus enabling visualization to highlight the nearest counterfactual datapoint (if a legit domain name is selected, then the nearest botnet DGA domain name will be shown, and vice versa). This tool will enable cybersecurity analysts to detect minimal changes in features' value to make the CTI system produce different classification results. Thus, CTI systems could gain more trust from users because they understand the explanation.

For the second opinion, we used two OSINT sources (Google Safe Browser and OTX AlienVault) [58]. We sent application programming interface (API) queries to these sources to retrieve a comment/report on the suspected domain in question. We fused this information with our botnet DGA model's output as a second opinion. The aggregate IoC from OSINT to confirm the AI/ML model's output and classification results can be submitted to OSINT repositories after thorough expert examinations to prevent submitting false/poor-quality findings. The submission of new IoC information to OSINT repositories needs to use extra caution, as it can intentionally or unintentionally be a way that can cause issues with other detectors relying on such OSINT and poison any training process. Therefore, computable CTI can advance OSINT communities for IoCs of new threats (Fig. 2).

## IV. RESULTS AND DISCUSSIONS

### A. COMPARING THE ACCURACY OF ML ALGORITHMS

The results of our experiments are shown in Tables 4 and 5. Overall, the random forest model achieved the highest accuracy, followed by the extra tree algorithm. Note that naive Bayes always showed the lowest performance among the compared algorithms. The highest accuracy (96.2%) was obtained using random forest with all seven features. The top-three essential features were the CharLength, ReputationAlexa, and TreeNewFeature.

We analyzed all features using statistical tests to select features with the strongest relationship with the output variable. Fig. 3 shows the results of the univariate selection Chi-squared test. The ReputationAlexa, CharLength, and TreeNewFeature features had the highest relationship with the class output. Then, we investigated how features are related to each other using a correlation matrix. As shown in Fig. 4, the TreeNewFeature, Char-Length, Entropy, and RE-Alexa features positively correlate with the output, and a negative correlation can be observed for the Min-RE-Botnets and IRad features. Furthermore, we performed feature importance analysis to score each feature in our proposed model. As shown in Fig. 5, the TreeNewFeature, ReputationAlexa, and CharLength features obtained the highest scores, which indicates that these features are essential for the output variable.

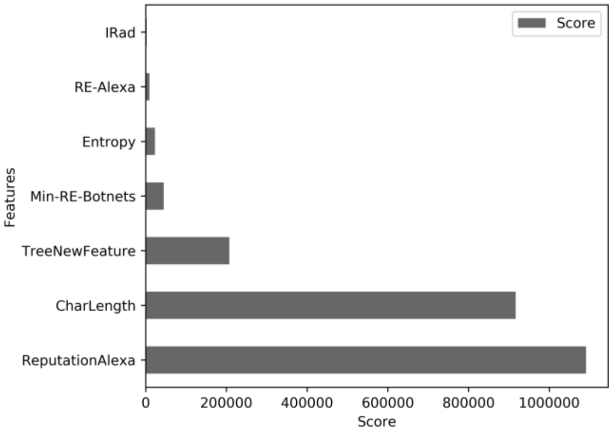


**FIGURE 3. Chi-squared test results.**

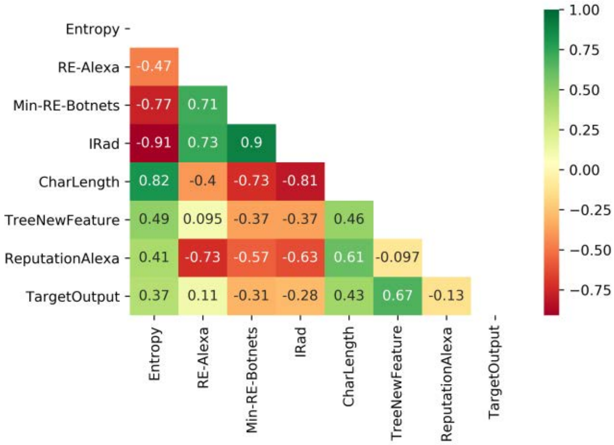


**FIGURE 4. Correlation matrix analysis results.**

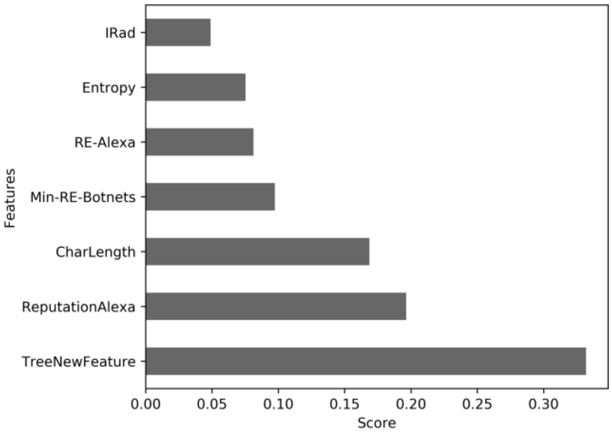


**FIGURE 5. Feature importance analysis results.**

**TABLE 4.** ML Model's accuracy using combinations of various features (1/2).

| Features | 3 features | 4 features | 4 features | 4 features | 4 features | 5 features | 5 features | 5 features |
|---|---|---|---|---|---|---|---|---|
| - CharLength | √ | √ | √ | √ | √ | √ | √ | √ |
| - TreeNewFeature | √ | √ | √ | √ | √ | √ | √ | √ |
| - ReputationAlexa | √ | √ | √ | √ | √ | √ | √ | √ |
| - RE-Alexa | - | √ | - | - | - | √ | √ | √ |
| - Min-RE-Botnets | - | - | √ | - | - | √ | - | - |
| - Entropy | - | - | - | √ | - | - | √ | - |
| - IRad | - | - | - | - | √ | - | - | √ |
| Logistic Regression | 89.5% | 89.8% | 89.5% | 90.6% | 90.0% | 90.1% | 90.7% | 90.0% |
| Random Forest | 92.7% | 94.6% | 95.1% | 94.4% | 94.8% | 95.7% | 95.3% | 95.2% |
| Naive Bayes | 83.2% | 83.5% | 82.7% | 82.5% | 82.5% | 82.9% | 82.9% | 82.9% |
| Extra Tree | 92.7% | 94.3% | 94.8% | 94.2% | 94.5% | 95.6% | 95.2% | 95.1% |
| Ensemble | 91.7% | 93.6% | 94.4% | 94.6% | 94.1% | 94.7% | 94.5% | 94.1% |

**TABLE 5.** ML Model's accuracy using combinations of various features (2/2).

| Features | 5 features | 5 features | 5 features | 6 features | 6 features | 6 features | 6 features | 7 features |
|---|---|---|---|---|---|---|---|---|
| - CharLength | √ | √ | √ | √ | √ | √ | √ | √ |
| - TreeNewFeature | √ | √ | √ | √ | √ | √ | √ | √ |
| - ReputationAlexa | √ | √ | √ | √ | √ | √ | √ | √ |
| - RE-Alexa | - | - | - | √ | √ | √ | - | √ |
| - Min-RE-Botnets | √ | √ | - | √ | √ | - | √ | √ |
| - Entropy | √ | - | √ | √ | - | √ | √ | √ |
| - IRad | - | √ | √ | - | √ | √ | √ | √ |
| Logistic Regression | 91.3% | 90.9% | 90.8% | 91.3% | 90.5% | 90.7% | 91.4% | 91.3% |
| Random Forest | 95.9% | 95.5% | 95.6% | 96.1% | 95.8% | 95.8% | 96.1% | 96.2% |
| Naive Bayes | 81.9% | 81.9% | 81.9% | 82.2% | 82.2% | 82.3% | 81.5% | 81.5% |
| Extra Tree | 95.8% | 95.4% | 95.5% | 96.1% | 95.7% | 95.8% | 96.1% | 96.2% |
| Ensemble | 95.2% | 94.6% | 94.8% | 95.2% | 94.6% | 94.6% | 95.1% | 95.0% |

### B. TIME COMPLEXITY TO CALCULATE THE FEATURES

ML classification that uses a statistical-based approach requires computations to calculate their features. Fig. 6 shows the computational cost of our approach in terms of time complexity to calculate the features needed in our model. Min-RE-Botnets and IRad require longer computation time than other features, as the equation (2) and (3) bring consequences of $O(n)$ linear time complexity, with $n =$ the number of DGA families (55 families in our experiments). This will become a disadvantage when the number of DGA families grows. However, the ReputationAlexa feature does not require heavy computations, as the preparations need to be done only once during the model training step: reading all the domains from Alexa and then learning the vocabulary dictionary of n-gram 3 to 5 characters to generate the weight matrix.

### C. COMPARISON WITH THE PREVIOUS WORK

Table 6 provides a comparison between our proposed random forest model and the previous work (Hoang and Vu [7]), which used the same approach (statistical features-based) and same random forest algorithm. Using the same datasets settings as in [7, pp. 7–8], for all experiments, our model gives a better detection rate (with an average of 98.9% accuracy), despite our approach using only seven features compared with their approach, which depends on 24 features.

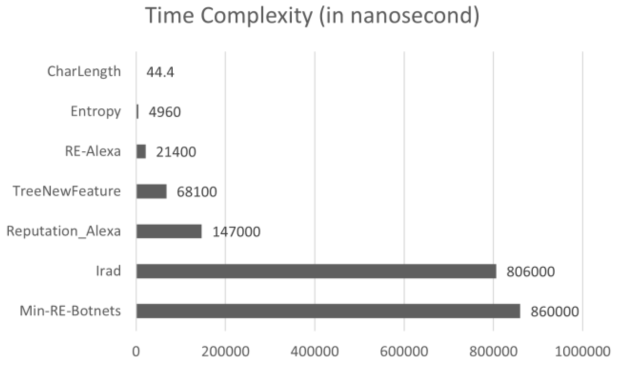


**FIGURE 6.** The computation time to prepare the features for our model.

These results give clear evidence on the advantage of the potential utilization of seven features proposed in our paper to deliver a satisfactory botnet DGA detection performance.

### D. ROBUSTNESS EVALUATION

First, we examined the performance of our random forest model with seven features using ground-truth datasets consisting of Alexa and 55 DGA families' domain names (1,998,502 domain names in total). As shown in Table 7, the character-based deep learning models produce slightly higher accuracy (∼99.0%) than our model (96.3% accuracy).

**TABLE 6. Comparison of the accuracy/detection rate.**

| Datasets | Our Random Forest Model (with only 7 features) | Hoang and Vu [7] (with 24 features) |
|---|---|---|
| 39 DGA families | 99.3% | 83.8% |
| 25 DGA families | 99.7% | 98.0% |
| 10 DGA families | 97.2% | 83.1% |
| 4 DGA families | 99.3% | 75.0% |
| *average =* | *98.9%* | *85.0%* |

**TABLE 7. Robustness against state-of-the-art DGA attacks [9]–[11].**

| Datasets | Our Model | Character-Based Deep Learning Model [8] | | | |
|---|---|---|---|---|---|
| | | Endgame [35] | CMU [36] | NYU [37] | MIT [38] |
| - Alexa Top 1M and 55 DGA families | 96.3% | 98.9% | 99.0% | 98.9% | 99.0% |
| - CharBot, MaskDGA, and DeepDGA attacks | 44.2% | 35.8% | 30.5% | 38.4% | 29.7% |
| - CharBot attack | 17.4% | 15.0% | 10.9% | 9.1% | 10.0% |
| - MaskDGA attack | 19.7% | 27.4% | 17.1% | 16.3% | 30.9% |
| - DeepDGA attack | 45.8% | 36.7% | 31.6% | 40.0% | 30.3% |

However, robustness evaluation with CharBot, MaskDGA, and DeepDGA attacks (394,000 domain names in total) give evidence that our model provides better defense against all the three DGA attacks (44,2% accuracy). Evaluation against individual DGA attacks shows that our model has better robustness against CharBot and DeepDGA attacks, except in the MaskDGA attack.

These results confirm the advantages of our model to be used for botnet DGA detection in dealing with harsh DGA attacks, in which a novel DGA attack can significantly drop the accuracy of a DGA classifier up to only 9.1% accuracy (in the case of the NYU model tested with CharBot attack). This tendency is similar to the previous works [9]–[11].

### *E. SHARING MECHANISM IN COMPUTABLE CTI*

We identified several protocols that potentials for implementing sharing AI/ML models, such as docker container, native (dependent on the tools used, e.g., joblib for Python); PMML/XML-based predictive model interchange format; and open neural network exchange (ONNX) the open standard for ML interoperability. We serialized our final ML model for botnet DGA detection using the ONNX and Python's joblib approaches. The serialization and deserialization could run smoothly. Although the trained model's file size could become large when the training data is enormous, sharing a trained/ready-to-use model is very convenient for others who need to analyze botnet DGA traffics without the burden of building and training a model.

### *F. BLENDING XAI AND OSINT*

Firstly, our CTI system displays a global explanation of the model (Fig. 7). Our model considers character length as a key feature in recognizing botnet DGA domain names. Therefore, the domain with a too-long number of characters tends to be a botnet DGA domain name, which is true, based on our ground-truth dataset. To trust the model, users must understand what the model is good at and when the model could go wrong. We provide visualization (Fig. 8), enabling a cybersecurity analyst to see where the classification is wrong, such as when a legit domain name exists, but the model classifies it as a botnet DGA domain name. Even though the model has high accuracy, if it classifies a well-known legit domain (such as google.com) as a botnet DGA, it is unacceptable.

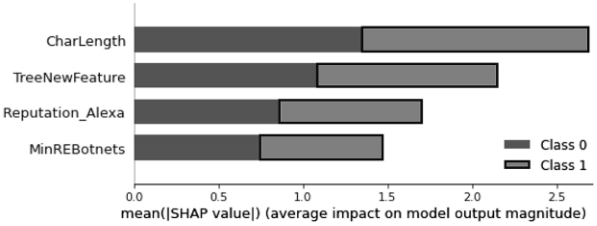
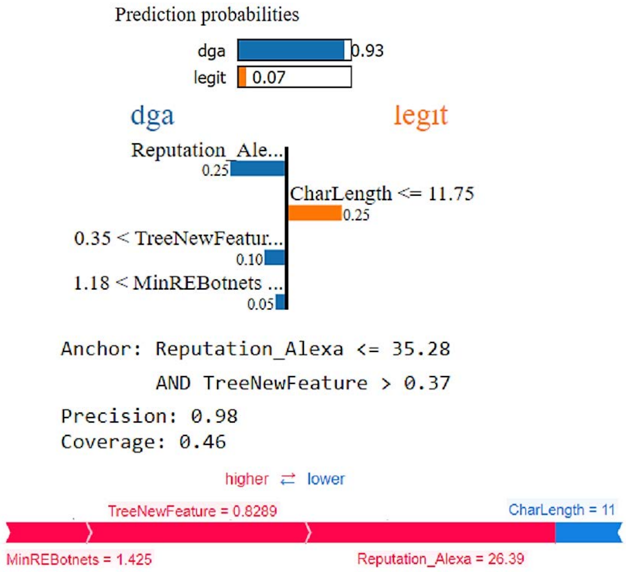


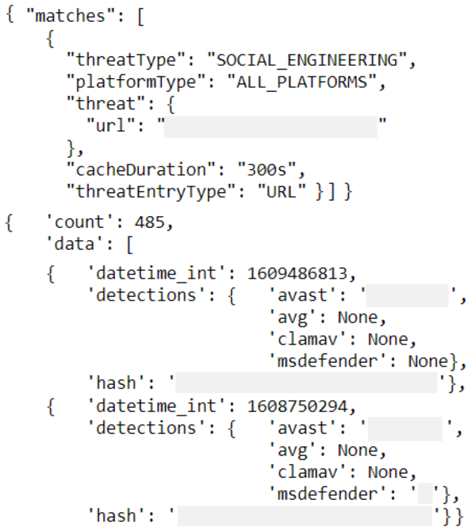
**FIGURE 7. SHAP global explanation summary plot.**



```
Anchor: Reputation_Alexa <= 35.28
        AND TreeNewFeature > 0.37
Precision: 0.98
Coverage: 0.46
```

**FIGURE 8. Local explanation: LIME shows prediction plot, ANCHORS display IF-THEN Rule, and SHAP provides a force plot.**

We provide local explanations or explain a single decision output to present the simplification idea of the logic of why the model produces that decision. We show an example of why the CTI system determines that a normal-looking domain name (of which the number of characters is not too excessive) is classified as a botnet DGA domain name (Fig. 8). LIME [54] shows easy-to-understand plots. Even though the short character length makes it look like a legit domain name, the reputation score calculation compared with Alexa Top 1M domain names causes the opposite decision. ANCHORS provides a similar explanation, but in the IF-THEN rule format [55]. Moreover, SHAP [53] displays how each feature's value is forcing a decision toward a legit or botnet DGA classification result. A counter explanation



```
{ "matches": [
    {
      "threatType": "SOCIAL_ENGINEERING",
      "platformType": "ALL_PLATFORMS",
      "threat": {
        "url": "[redacted]"
      },
      "cacheDuration": "300s",
      "threatEntryType": "URL" }] }
{    'count': 485,
     'data': [
     {    'datetime_int': 1609486813,
          'detections': {    'avast': '[redacted]',
                             'avg': None,
                             'clamav': None,
                             'msdefender': None},
          'hash': '[redacted]'},
     {    'datetime_int': 1608750294,
          'detections': {    'avast': '[redacted]',
                             'avg': None,
                             'clamav': None,
                             'msdefender': '[redacted]'},
          'hash': '[redacted]'}}
```

**FIGURE 9. A second opinion from blending XAI and OSINT: integrating API query results from Google Safe Browser and OTX AlienVault.**

is helpful to make cybersecurity analysts trust our model by providing examples: two domain names with similar characteristics but are classified as different classes; one is a botnet DGA and the other a legit domain name.

After showing SHAP, LIME, ANCHORS, and counterfactual explanations, we continue providing a second opinion (Fig. 9) by integrating API query results from two OSINT sources (Google Safe Browser and OTX AlienVault). Therefore, we confirm that a good explanation and second opinion (by implementing XAI and OSINT blend) are keys to establishing trust in using the shared AI/ML model.

In our study of botnet DGA detection, automation bias refers to an act where a cybersecurity analyst never doubts the AI/ML model's decision output, whatever they are. Such as when the model falsely detects a domain name as a malicious botnet DGA, and the cybersecurity analyst trusts it too much. We emphasize that blending XAI and OSINT could solve the automation bias through a second opinion.

Cyber false flags are hackers' tactics to deceive or misguide attribution attempts and covert cyberattacks [59]. By blending XAI and OSINT into AI/ML-based CTI systems, cybersecurity analysts have a handy tool to compare any information from OSINT sources, with the model's results taken from CTI-sharing repositories (measure twice, cut once using the AI/ML model to confirm OSINT information). This description highlights the usefulness of our proposed XAI and OSINT blend for cyber false-flag phenomena.

### G. PRACTICAL IMPLICATIONS OF COMPUTABLE CTI

Reducing human intervention in cybersecurity decision-making using AI/ML automation will help security analysts in security operations center environments to win the arms races against new cyber threats. The computable CTI paradigm emphasizes a robust AI/ML model with adversarial defense techniques, also blending XAI and OSINT to solve the automation bias. For example, in our botnet DGA case study, OSINT data become a second opinion (or validation) for known DGA domain names. Thus, we achieved cybersecurity decision-making automation. When no information in the OSINT database exists regarding a suspected domain name, security analysts can still make a fair decision by referencing the explanations produced by XAI techniques.

Computable CTI paradigm also encourages cybersecurity communities to contribute their carefully curated CTI detection outputs to enrich IoC data in OSINT repositories. OSINT APIs integrate AI/ML models to enable submitting new threat information to the OSINT database. In our case study of botnet DGA detection, we used the OTX AlienVault's DirectConnect API to demonstrate submitting new confirmed and validated findings, when no available OSINT exists for the botnet DGA domain names. Therefore, computable CTI implies two-way interactions: gaining benefits from aggregating OSINT threat data and contributing to the latest threats' IoCs for tackling new global attack vectors.

Recently, we have been observing the emergence of public repositories/marketplaces for ready-to-use AI/ML models, such as in TensorFlow Hub. Various models for common problem domains (image, text, video, and audio) are available to be used for transfer learning; however, AI/ML models for CTI applications are scarce [60], [61]. Our study is a gateway for future AI/ML model-based CTI-sharing research. Therefore, in this section, we elaborate on the frameworks needed in computable CTI (Fig. 2) to ensure that cybersecurity communities will be encouraged to share their AI/ML models for CTI sharing.

First, regarding the interoperability of the AI/ML model, we demonstrated how we could manage interoperability when sharing the AI/ML model using ONNX. Adopting this standard removes the barrier of being locked on one AI/ML platform. Sharing CTI models in the ONNX standard will reach a wider audience of cybersecurity communities.

Second, users' privacy must be protected because model sharing takes place among users. We propose adopting privacy labels (color-coded: white, green, amber, and red) relating to privacy-related measures and compliance with privacy regulations [62] on the shared models. Various privacy-preserving techniques can be employed when the models include storing, processing, and transferring private information [15], [16].

Third, the computable CTI paradigm encourages adopting the code-signing practice to ensure the integrity and authenticity of the shared AI/ML model. Sigstore, a recently announced project of The Linux Foundation aiming to foster adopting cryptographic signing, might become a catalyst for the wide adoption of computable CTI in cybersecurity and open-source communities.

## V. CONCLUSION

First, we showcase a novel model for botnet DGA detection. Our random forest model achieved 96.3% accuracy (tested with datasets of 55 botnet DGA families) and outperformed the previous work (see Section IV.C). Our model is also more robust against three state-of-the-art DGA adversarial attacks (MaskDGA, CharBot, and DeepDGA) than the previous works (see Section IV.D).

Second, we highlight the practicality of blending XAI and OSINT to deliver better AI explainability through second opinion approaches, thus mimicking the second opinion phenomena in hospital/medical situations to confirm the results/findings. We advocate the XAI and OSINT as an antidote for skepticism toward the model's output, which might contribute to the CTI system's trust and prevent automation bias when users have too much trust in the CTI system's output. Blending XAI and OSINT also has a potential for solving the false-flag problems.

Third, we underline the case study of botnet DGA detection with XAI and OSINT blend as evidence to validate our proposed computable CTI paradigm. Improving trust might result in a paradigm-shift phenomenon. Cybersecurity communities will leave the traditional CTI-sharing paradigm (sharing only threat indicators, such as threat domain names), and communities will start to share AI/ML models for CTI systems. With the emergence of the computable CTI-sharing paradigm, additional collaboration among cybersecurity communities will occur to develop advanced AI/ML-based CTI systems. For instance, using transfer-learning techniques to develop new AI/ML for new cybersecurity tasks/problems utilizing the shared models.

The limitations of our DGA detection model are the time complexity when calculating the features (Section IV.B) and the limited robustness against MaskDGA attacks (Section IV.D). Future improvement should focus on crafting better features and adversarial defense strategies. Moving target defense (MTD) [63] can potentially raise the model's robustness by combining various models to work together.

## GLOSSARY

| | |
|---|---|
| CharLength | domain name character length |
| CTI | cyber threat intelligence |
| DGA | domain generation algorithm |
| IoC | indicators of compromise |
| IRad | information radius |
| Min-RE-Botnets | minimum of relative entropy botnets |
| OSINT | open-source intelligence |
| RE | relative entropy |
| RE-Alexa | the RE of a domain name to Alexa |
| ReputationAlexa | Alexa reputation score |
| TreeNewFeature | a new feature generated by decision tree |
| XAI | explainable artificial intelligence |

## ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, an associate editor, and a reviewers for their constructive and generous feedbacks. The authors thank all the professors and colleagues from the Institut Teknologi Sepuluh Nopember. Also, they thank Jonathan Peck for making the CharBot code available to them.

## REFERENCES

[1] F. Skopik, G. Settanni, and R. Fiedler, "A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing," *Comput. Secur.*, vol. 60, pp. 154–176, Jul. 2016, doi: 10.1016/j.cose.2016.04.003.

[2] R. Confalonieri, L. Coba, B. Wagner, and T. R. Besold, "A historical perspective of explainable artificial intelligence," *WIREs Data Mining Knowl. Discovery*, vol. 11, no. 1, p. e1391, Jan. 2021, doi: 10.1002/widm.1391.

[3] M.-A. Clinciu and H. Hastie, "A survey of explainable AI terminology," in *Proc. 1st Workshop Interact. Natural Lang. Technol. Explainable Artif. Intell. (NLXAI)*, 2019, pp. 8–13, doi: 10.18653/v1/W19-8403.

[4] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*. Cham, Switzerland: Springer, 2018, pp. 3–15, doi: 10.1007/978-3-319-91473-2_1.

[5] M. Singh, M. Singh, and S. Kaur, "Issues and challenges in DNS based botnet detection: A survey," *Comput. Secur.*, vol. 86, pp. 28–52, Sep. 2019, doi: 10.1016/j.cose.2019.05.019.

[6] T. S. Wang, H.-T. Lin, W.-T. Cheng, and C.-Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," *Comput. Secur.*, vol. 64, pp. 1–15, Jan. 2017, doi: 10.1016/j.cose.2016.10.001.

[7] X. D. Hoang and X. H. Vu, "An improved model for detecting DGA botnets using random forest algorithm," *Inf. Secur. J., Global Perspective*, pp. 1–10, Jun. 2021, doi: 10.1080/19393555.2021.1934198.

[8] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489147.

[9] L. Sidi, A. Nadler, and A. Shabtai, "MaskDGA: An evasion attack against DGA classifiers and adversarial defenses," *IEEE Access*, vol. 8, pp. 161580–161592, 2020, doi: 10.1109/ACCESS.2020.3020964.

[10] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. Olumofin, B. Yu, A. Nascimento, and M. De Cock, "CharBot: A simple and effective method for evading DGA classifiers," *IEEE Access*, vol. 7, pp. 91759–91771, 2019, doi: 10.1109/ACCESS.2019.2927075.

[11] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-tuned domain generation and detection," in *Proc. ACM Workshop Artif. Intell. Secur.*, Vienna, Austria, Oct. 2016, pp. 13–21, doi: 10.1145/2996758.2996767.

[12] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Comput. Secur.*, vol. 87, Nov. 2019, Art. no. 101589, doi: 10.1016/j.cose.2019.101589.

[13] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Comput. Secur.*, vol. 72, pp. 212–233, Jan. 2018, doi: 10.1016/j.cose.2017.09.001.

[14] P. Pawlinski, P. Jaroszewski, P. Kijewski, L. Siewierski, P. Jacewicz, P. Zielony, and R. Zuber, "Actionable information for security incident response," in *Proc. Eur. Union Agency Netw. Inf. Secur.*, Heraklion, Greece, 2014, pp. 1–68.

[15] H. C. Tanuwidjaja, R. Choi, S. Baek, and K. Kim, "Privacy-preserving deep learning on machine learning as a service—A comprehensive survey," *IEEE Access*, vol. 8, pp. 167425–167447, 2020, doi: 10.1109/ACCESS.2020.3023084.

[16] A. Boulemtafes, A. Derhab, and Y. Challal, "A review of privacy-preserving techniques for deep learning," *Neurocomputing*, vol. 384, pp. 21–45, Apr. 2020, doi: 10.1016/j.neucom.2019.11.041.

[17] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.

[18] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *Proc. 11th Int. Conf. Cyber Conflict (CyCon)*, May 2019, pp. 1–18, doi: 10.23919/CYCON.2019.8756865.

[19] S. Gong, J. Cho, and C. Lee, "A reliability comparison method for OSINT validity analysis," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5428–5435, Dec. 2018, doi: 10.1109/TII.2018.2857213.

[20] MITRE. (2021). *Virus Total Data Poisoning Case Studies*. [Online]. Available: http://git- hub.com/mitre/advmlthreatmatrix/blob/master/pages/case-studies-page.md#virustotal-poisoning

[21] T. D. Wagner, E. Palomar, K. Mahbub, and A. E. Abdallah, "A novel trust taxonomy for shared cyber threat intelligence," *Secur. Commun. Netw.*, vol. 2018, Jun. 2018, Art. no. e9634507, doi: 10.1155/2018/9634507.

[22] M. Nourani, S. Kabir, S. Mohseni, and E. D. Ragan, "The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems," in *Proc. AAAI Conf. Hum. Comput. Crowdsourcing*, 2019, vol. 7, no. 1, pp. 97–105.

[23] R. Goebel, A. Chander, K. Holzinger, F. Lecue, Z. Akata, S. Stumpf, P. Kieseberg, and A. Holzinger, "Explainable AI: The new 42?" in *Machine Learning and Knowledge Extraction*. Cham, Switzerland: Springer, 2018, pp. 295–303, doi: 10.1007/978-3-319-99740-7_21.

[24] W. Samek and R. K. Müeller, "Towards explainable artificial intelligence," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Lecture Notes in Computer Science), vol. 11700. Cham, Switzerland: Springer, 2019, pp. 5–22, doi: 10.1007/978-3-030-28954-6_1.

[25] L. Longo, R. Goebel, F. Lecue, P. Kieseberg, and A. Holzinger, "Explainable artificial intelligence: Concepts, applications, research challenges and visions," in *Machine Learning and Knowledge Extraction*. Cham, Switzerland: Springer, 2020, pp. 1–16, doi: 10.1007/978-3-030-57321-8_1.

[26] S. Hepenstal and D. McNeish, "Explainable artificial intelligence: What do you need to know?" in *Augmented Cognition. Theoretical and Technological Approaches*. Cham, Switzerland: Springer, 2020, pp. 266–275, doi: 10.1007/978-3-030-50353-6_20.

[27] A. Ignatiev, "Towards trustable explainable AI," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, vol. 5, Jul. 2020, pp. 5154–5158, doi: 10.24963/ijcai.2020/726.

[28] J. M. Schoenborn and K. D. Althoff, "Recent trends in XAI: A broad overview on current approaches, methodologies and interactions," in *Proc. ICCBR Workshops*, vol. 2567, 2019, pp. 51–60.

[29] H. Suryotrisongko, "Botnet DGA detection," *IEEE Code Ocean*, Jun. 2021, doi: 10.24433/CO.4005597.v2.

[30] H. Suryotrisongko and Y. Musashi, "Botnet DGA dataset," *IEEE Dataport*, May 2020, doi: 10.21227/rg6z-z622.

[31] A. Satoh, Y. Nakamura, D. Nobayashi, K. Sasai, G. Kitagata, and T. Ikenaga, "Clustering malicious DNS queries for blacklist-based detection," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 7, pp. 1404–1407, Jul. 2019, doi: 10.1587/transinf.2018EDL8211.

[32] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *Research in Attacks, Intrusions and Defenses*. Cham, Switzerland: Springer, 2014, pp. 1–21, doi: 10.1007/978-3-319-11379-1_1.

[33] K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," *Neural Comput. Appl.*, vol. 28, no. 7, pp. 1541–1558, Jul. 2017, doi: 10.1007/s00521-015-2128-0.

[34] 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. 67:1–67:36, Jul. 2018, doi: 10.1145/3191329.

[35] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," Nov. 2016, *arXiv:1611.00791*. Accessed: Dec. 30, 2021.

[36] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W. Cohen, "Tweet2Vec: Character-based distributed representations for social media," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, vol. 2, Aug. 2016, pp. 269–274, doi: 10.18653/v1/P16-2044.

[37] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 649–657.

[38] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2016, pp. 1041–1044, doi: 10.1145/2911451.2914762.

[39] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 0210–0215, doi: 10.23919/MIPRO.2018.8400040.

[40] C. Meske, E. Bunde, J. Schneider, and M. Gersch, "Explainable artificial intelligence: Objectives, stakeholders, and future research opportunities," *Inf. Syst. Manage.*, vol. 39, no. 1, pp. 53–63, Jan. 2022, doi: 10.1080/10580530.2020.1849465.

[41] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[42] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[43] S. M. Mathews, "Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review," in *Intelligent Computing*. Cham, Switzerland: Springer, 2019, pp. 1269–1292, doi: 10.1007/978-3-030-22868-2_90.

[44] W. Pieters, "Explanation and trust: What to tell the user in security and AI?" *Ethics Inf. Technol.*, vol. 13, no. 1, pp. 53–64, Mar. 2011, doi: 10.1007/s10676-010-9253-3.

[45] M. Stevanovic, J. M. Pedersen, A. D'Alconzo, S. Ruehrup, and A. Berger, "On the ground truth problem of malicious DNS traffic analysis," *Comput. Secur.*, vol. 55, pp. 142–158, Nov. 2015, doi: 10.1016/j.cose.2015.09.004.

[46] C. Patsakis, F. Casino, and V. Katos, "Encrypted and covert DNS queries for botnets: Challenges and countermeasures," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101614, doi: 10.1016/j.cose.2019.101614.

[47] *Netlab OpenData Project*. Accessed: Dec. 30, 2021. [Online]. Available: http://data.netlab.360.com/

[48] Y. Matsubara, Y. Musashi, K. Sugitani, and T. Moriyama, "Open DNS resolver activity in campus network system," in *Proc. 8th Int. Conf. Intell. Netw. Intell. Syst. (ICINIS)*, Nov. 2015, pp. 145–148, doi: 10.1109/ICINIS.2015.30.

[49] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1663–1677, Oct. 2012, doi: 10.1109/TNET.2012.2184552.

[50] R. Sharifnya and M. Abadi, "DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic," *Digital Invest.*, vol. 12, pp. 15–26, Mar. 2015, doi: 10.1016/j.diin.2014.11.001.

[51] H. Zhao, Z. Chang, G. Bao, and X. Zeng, "Malicious domain names detection algorithm based on N-gram," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–9, Feb. 2019, doi: 10.1155/2019/4612474.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.

[53] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 4765–4774.

[54] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[55] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1527–1535.

[56] J. Klaise, A. Van Looveren, G. Vacanti, and A. Coca. (2020). *Alibi: Algorithms for Monitoring and Explaining Machine Learning Models*. [Online]. Available: https://github.com/SeldonIO/alibi

[57] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson, "The what-if tool: Interactive probing of machine learning models," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 56–65, Jan. 2020, doi: 10.1109/TVCG.2019.2934619.

[58] C. Sauerwein, I. Pekaric, M. Felderer, and R. Breu, "An analysis and classification of public information security data sources used in research and practice," *Comput. Secur.*, vol. 82, pp. 140–155, May 2019, doi: 10.1016/j.cose.2018.12.011.

[59] F. Skopik and T. Pahi, "Under false flag: Using technical artifacts for cyber attack attribution," *Cybersecurity*, vol. 3, no. 1, p. 8, Mar. 2020, doi: 10.1186/s42400-020-00048-4.

[60] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020, doi: 10.1109/TAI.2021.3054609.

[61] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.

[62] A. Albakri, E. Boiten, and R. De Lemos, "Sharing cyber threat intelligence under the general data protection regulation," in *Privacy Technologies and Policy*, Cham, Switzerland: Springer, 2019, pp. 28–41, doi: 10.1007/978-3-030-21752-5_3.

[63] R. Izmailov, P. Lin, S. Venkatesan, and S. Sugrim, "Combinatorial boosting of classifiers for moving target defense against adversarial evasion attacks," in *Proc. 8th ACM Workshop Moving Target Defense*, Nov. 2021, pp. 13–21, doi: 10.1145/3474370.3485661.

**HATMA SURYOTRISONGKO** (Member, IEEE) was born in Yogyakarta, Indonesia, in 1984. He received the Bachelor of Computer Science degree from the Universitas Gadjah Mada, Indonesia, in 2007, and the master's degree in knowledge-based information engineering from the Toyohashi University of Technology, Japan, in 2011. He is currently pursuing the Ph.D. degree with the Graduate School of Science and Technology, Kumamoto University, Japan.

Since 2014, he has been a Lecturer with the Cybersecurity and Smart City Laboratory, Information Technology Department, Faculty of Intelligent Electrical and Informatics Technology (F-ELECTICS), Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. His research interests include cybersecurity, cyber threat intelligence, artificial intelligence (AI), explainable AI, confidential/privacy-preserving machine learning (ML), adversarial ML, quantum ML, and post-quantum cryptography.

Mr. Suryotrisongko is a member of the IEEE Computer Society, the Information Processing Society of Japan (IPSJ), and an Association for Computing Machinery (ACM).

**YASUO MUSASHI** (Member, IEEE) had been with the Information Processing Center, Kumamoto University, as a Cooperative Researcher and an Assistant Professor, in 1997. Since 2002, he had been as an Associate Professor at the Center for Multimedia and Information Technologies. He was a Guest Scientist of the Johann Wolvgang Goethe Universitaet Frankfurt am Main a half year of 2005, from January 2005 to July 2005. Since May 2014, he had been with the Center for Management and Information Technologies (CMIT), Kumamoto University. He has been a Full Professor at CMIT, since 2015. His current research interests include computer network security and developing security incident detection and prevention systems.

**AKIO TSUNEDA** (Member, IEEE) was born in Nagasaki, Japan, in 1967. He received the B.E., M.E., and D.E. degrees in computer science and communication engineering from Kyushu University, Fukuoka, Japan, in 1990, 1992, and 1995, respectively.

From 1995 to 1996, he was with the Department of Computer Science and Communication Engineering, Kyushu University. During 2003–2004, he spent eight months at the University of California at Berkeley and two months at the University of Birmingham as a Visiting Scholar. He is currently a Professor with the Faculty of Advanced Science and Technology, Kumamoto University, Japan. His research interests include nonlinear dynamical systems, chaos circuits, pseudorandom sequences, and digital communications.

Dr. Tsuneda is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE) and the Institute of Electrical Engineers of Japan (IEEJ).

**KENICHI SUGITANI** is currently a Professor at the Graduate School of Science and Technology (GSST), Kumamoto University. He is also the Director of the Center for Management of Information and Technologies (CMIT), Kumamoto University. His research interests include web technology and computer network security.