International Journal of Information Security
https://doi.org/10.1007/s10207-023-00686-y

REGULAR CONTRIBUTION

Malicious website identification using design attribute learning

Or Naim[1] · Doron Cohen[1] · Irad Ben-Gal[1]

Accepted: 13 March 2023

Abstract

Malicious websites pose a challenging cybersecurity threat. Traditional tools for detecting malicious websites rely heavily on industry-specific domain knowledge, are maintained by large-scale research operations, and result in a never-ending attacker–defender dynamic. Malicious websites need to balance two opposing requirements to successfully function: escaping malware detection tools while attracting visitors. This fundamental conflict can be leveraged to create a robust and sustainable detection approach based on the extraction, analysis, and learning of design attributes for malicious website identification. In this paper, we propose a next-generation algorithm for extended design attribute learning that learns and analyzes web page structures, content, appearances, and reputation to detect malicious websites. Results from a large-scale experiment that was conducted on more than 35,000 websites suggest that the proposed algorithm effectively detects more than 83% of all malicious websites while maintaining a low false-positive rate of 2%. In addition, the proposed method can incorporate user feedback and flag new suspicious websites and thus can be effective against zero-day attacks.

Keywords Malicious websites · Website design attributes · Machine learning · Cybersecurity · Human–computer interaction

1 Introduction

Malicious websites form a major cyberattack vector [1]. Detecting malicious websites is a challenging task, as malicious websites come in different formats and are often bundled with useful content, such as software, that is downloaded by naive users.

Traditional detection techniques rely on domain expertise and leverage advanced industry knowledge to detect indications of malicious activity. This approach results in a constant need for the research and development of detection capabilities, such as signatures and tailor-made features, to detect malicious activity. Among these techniques, we can find Document Object Model (DOM) analysis methods [2–4], JavaScript scanning techniques [5, 6], analyses of the software properties linked to websites [7–10], URL analysis approaches [11–14], user navigation path analysis methods [15, 16], text mining [17–19], and process mining strategies [20].

One problematic effect related to the domain expertise approach is the unending arms race that it creates. The tailor-made features designed as part of the detection method will eventually be bypassed by the attacker, resulting in a need to create new tailor-made features. In addition, this approach is mainly relevant for detecting known threats and attack vectors; as a result, it seems to be far less effective for detecting emerging threats and zero-day attacks.

Another problematic effect of the traditional detection techniques is the symmetry they create; the attacker can essentially have access to the same data that the defender uses to train their detection model, reverse engineer it, and bypass the model.

As previously suggested, malicious websites need to balance two opposing requirements to successfully function: escaping detection tools while attracting visitors [21, 22]. To attract visitors, a website needs to signal its claimed functionality to potential users by leveraging appearance, content, and experience [23]. This fundamental conflict can be exploited to create a robust and sustainable classification approach. As suggested in a previous work by Cohen et al. [24], websites can be accurately classified and categorized by their design attributes.

In this paper, we propose a framework for detecting malicious websites by extensively learning their design attributes.

✉ Or Naim
ornaim@mail.tau.ac.il

[1] Tel Aviv University, Tel Aviv, Israel

Published online: 24 March 2023

The suggested approach was tested on a large-scale imbalanced dataset that included a total of 35,707 website records, 697 of which were malicious. This dataset was assembled to accurately represent the commercial, real-life scenario of malicious website identification. Much attention was given to properly representing the malicious website ratio out of the entire population [25] and to ensuring that the malicious websites were generated from the same initial list and ranking system as the legitimate website population [26].

Validating the suggested approach on a real-life large-scale dataset poses some key challenges. For instance, the noise and variance are much greater than those of a carefully selected dataset. In addition, an extremely imbalanced dataset requires proper measures when analyzing the data and training the relevant models.

The suggested approach can effectively detect more than 83% of malicious websites while maintaining a low false-positive rate (FPR) of 2%. In addition, it was proved effective in detecting malicious and suspicious websites that allegedly slipped under the radar in previous studies. The suggested framework also offers explainability and can leverage the cybersecurity practitioner's experience and feedback to perform better and respond to emerging threats.

Another part of our contribution lies in assembling and sharing this unique and high-quality dataset, consisting of multiple design attribute features and third-party enrichment, with the research community.

This paper is divided into several sections. First, we review prior research related to this field in Sect. 2. Next, we describe our methodology, including the dataset formulation and the feature extraction in Sect. 3. Then, we present and explain our results in Sect. 4. Finally, we discuss the implications of our findings in "Conclusion" section.

2 Related work

Malicious website detection techniques can traditionally be divided into 2 main approaches: dynamic analysis and static analysis [27].

Dynamic analysis is usually performed by analyzing a website's execution dynamics [6, 28–30]. The basis of this approach involves the idea of looking for a signature of malicious activity, such as the creation of an unusual process, repeated redirection, etc. Dynamic analysis techniques have inherent risks and are difficult to implement and generalize. These techniques are often implemented in controlled or isolated environments [5] using virtual machines [31] or honey client systems [32]. However, this type of approach provides deeper visibility into website behavior, as features extracted using dynamic analysis can accurately capture processes and contents that are available only after the website is fully loaded.

Static analysis focuses on the content and information that are available without executing the website's actual source code [11, 33]. The extracted features can typically include lexical features from the URL string [12, 13, 34, 35], HTML and JavaScript content, information about the host and the domain, and traffic and usage intelligence provided by third parties. Static analysis techniques that apply machine learning have been extensively investigated and have achieved good results.

Static content analysis has been found to be highly effective in detecting phishing websites. Under the assumption that a phishing website aims to lure the end user to enter their credentials and sensitive information, a limited set of domain expert features can be extracted from URL strings and HTML elements to accurately detect phishing websites. HTML elements, such as <ifrmae> or <input> tags, accompanied by indicative words, such as "password" and "credit card," were previously suggested to be highly effective in detecting phishing websites [36, 37]. Additional expert-based features, such as the number of anchors and links, were also investigated [38, 39], and when combined with previous work, the authors achieved a true-positive rate (TPR) of 95% for the specific scenario of phishing website detection. In an attempt to create a more robust detection technique, Altay suggested a keyword density-based approach for detecting malicious websites [40].

This technique was tested with a support vector machine (SVM) model that was trained on a large-scale dataset, and it achieved a high accuracy of 96.7% and a TPR of 94.2%. Similar approaches were suggested for successfully detecting click hijacking attacks on web pages [41, 42]. These approaches are well suited for detecting websites involved in phishing attacks, which limits their ability to detect other types of malicious websites.

In an attempt to generalize static web page content analysis to detect different types of malicious websites, Amrutkar proposed the kAYO approach, which combines static analyses of mobile web pages based on URL, JS, HTML, and mobile-specific contents [43]. A logistic regression model was trained on a large-scale imbalanced dataset and achieved a high TPR of 89%. However, the overall accuracy was only 90%, and the FPR was 8%. McGahagan performed a comprehensive evaluation of web page content for detecting malicious websites via 8 different supervised machine learning models and reported an accuracy of 89% with an FPR that could reach 10% [44]. This work emphasized both the potential and the challenge of using static analysis for detecting non-phishing malicious websites and raised concerns regarding the ability to implement this approach in a real-life commercial scenario due to the high induced FPR.

To better understand the implications for a real-life commercial scenario, one can examine a mid-market enterprise in the United States (US). The average US enterprise employs

Table 1 Comparison among the results of existing approaches that aim to detect a wide range of malicious websites

Related work	Year	Problem scope	DB size (K)	Imbalance level	Malicious prior probability (%)	ACC (%)	TPR (%)	FPR (%)
Amrutkar—kAYO	2017	Static analysis of mobile web pages	350	Extremely Imbalanced	1.5	90	89	8
Mcgahagan—A Comprehensive Evaluation of Webpage Content	2019	Malicious websites detection	40	Imbalanced	14.5	89	N/A	10
Liu and Lee—CNN Based Malicious Website Detection	2020	Malicious websites detection	6	Balanced	38.5	N/A	93	5.3
Cohen – Design attributes learning	2021	Malicious websites detection	15	Highly Imbalanced	3.5	98	66	< 1

between 1000 and 2000 people. In 2021, the average US internet user was accessing more than 100 unique web pages every day. As a result, we can estimate that altogether, the employees of one mid-market enterprise are accessing at least 100,000 web pages every day. Under the careful assumption that at least 10,000 of these websites are unique, an FPR of 8% means that a system will provide alerts regarding more than 800 web pages on a daily basis.

Liu suggested that both the lower accuracy and high FPR achieved by static analysis techniques in recent years are the result of the spam techniques used by malicious websites [45]. These techniques cause meaningful content to be invisible to static analysis tools. As a result, a convolutional neural network (CNN) model that analyzes captured website images was suggested. This model was trained on a balanced dataset containing 6K screenshots and reported a TPR of 93.6% and an FPR of 5.3%.

As suggested by Singh and Goyal [46], machine learning techniques to detect malicious websites heavily rely on extracting, learning, and selecting the relevant attributes. In their comparison, it was found that previous models that rely on attribute learning for malicious website identification have restricted themselves to a limited set of attributes, mainly due to computational limitations. It is also suggested that using a holistic approach will optimize attribute learning and classification results.

Cohen et al. [24] proposed a wide website assessment scheme based on the website design features (visual and non-visual features) contained on a web page and their related features. The algorithm implemented by Coen et al. utilized the web page URL, HTML, DOM, and CSS for website classification and achieved a classification accuracy of over 90%. Table 1 presents a comparison of the results yielded by the existing approaches that aim to detect a wide range of malicious websites.

The algorithm developed herein extends Cohen et al.'s work by enhancing the assessment scheme with the end-user observation standpoint by capturing a complete screenshot of each web page, analyzing its color coding and performing object detection. In addition, the suggested algorithm parses 3rd-party metrics regarding web page performance and examines them thoroughly for conducting malicious website identification on a real-life large-scale dataset.

3 Methodology

The suggested approach aims to detect malicious websites in general by using binary classification models and is not limited to specific attack vectors or techniques. The principal formula for binary classification represents the probability of a specific URL being malicious based on a set of features and a set of parameters learned during model training, as shown in Eq. 1.

$$P(y = 1|x; \theta) \tag{1}$$

where y-classification of a URL $\in$ {1-malicious,0-legitimate}, x-set of features, θ-set of parameters and or weights learned during training

The cornerstone of this work is to leverage the built-in tradeoff that malicious websites must balance: escaping malware detection tools while attracting visitors. This conflict manifests in a considerable way when examining malicious website design attributes and comparing them to those of a legitimate website.

In terms of design attributes, we refer to all visual and non-visual elements that a web page consists of [24]. Among these attributes, we can find HTML code and hierarchies, JavaScript, CSS, color tables, styles, font types, objects, etc. [47–50]. In addition, we also refer to the actual appearance of the website once its content is loaded and rendered. As a result, the suggested approach is a hybrid technique that enhances the static analysis method with aspects of dynamic

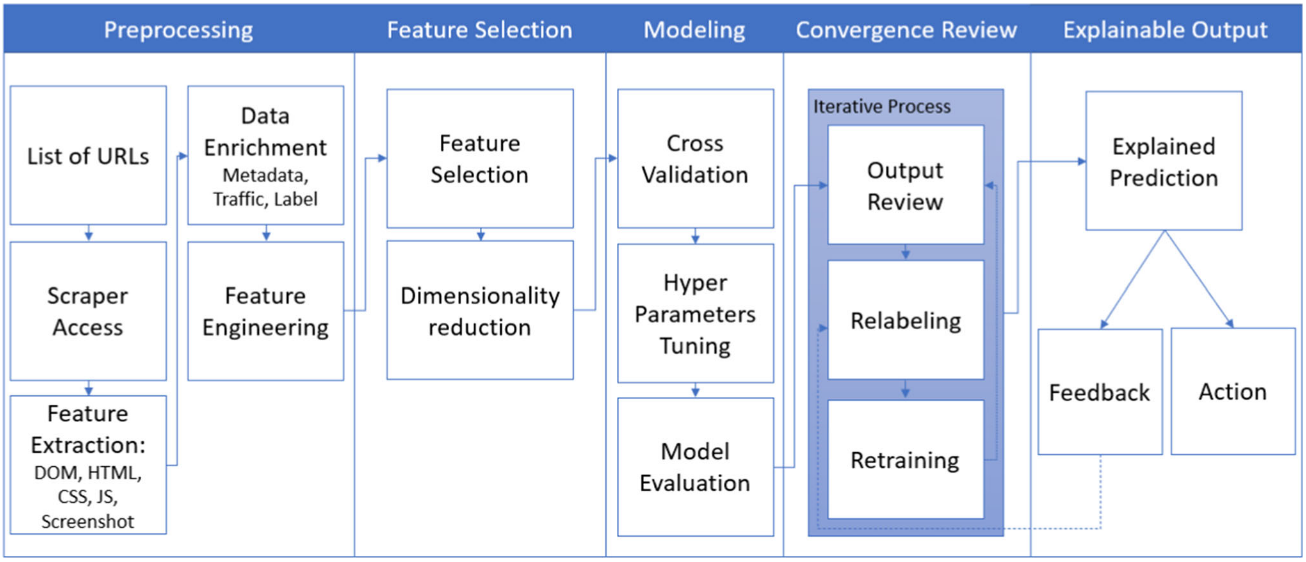


Fig. 1 Flow diagram of the proposed framework

analysis to capture and represent the actual content and appearance of a website.

By extracting design attributes from both the website's source code and the end-user observation standpoint, the suggested approach enables the identification of hidden patterns and mismatches between what exists behind the scenes and what is actually displayed on the screen.

The suggested approach consists of five main pillars, as demonstrated in Fig. 1: preprocessing, feature selection, modeling, convergence review, and providing explainable output. A main source of novelty of the suggested approach is at the stage of the preprocessing pillar that is explained in the following subsections. As part of the preprocessing stage, a dataset of 35,707 website records was created. Each URL was accessed by an automated scraper to extract the design attributes and enrich the website's data with a 3rd-party data. Feature selection and dimensionality reduction were then applied according to the relevant trained model. After the model was effectively tuned and evaluated, a convergence experiment was performed to simulate the interaction between the model prediction and a security practitioner that accordingly takes action. The model output includes explainability, which enables better interpretation of the obtained prediction and allows the user to provide feedback that tunes the model according to his or her preferences.

3.1 Dataset formation

To propose a robust and sustainable classification approach for malicious websites, any method must be evaluated on a high-quality large-scale website dataset that accurately demonstrates the high variance in the internet space and represents the low prior probability of actually being a malicious website based on the low percentage of malicious websites in the real world [25].

Assembling such a dataset is a complex operation that consists of three main phases: assembling an appropriately labeled list of websites, extracting relevant features from each website, and enriching the extracted features with third-party data sources.

The operation of assembling a labeled list of websites often starts by creating a large sample of websites using an external ranking system to capture a list of top-ranked websites. While capturing a non-skewed sample that correctly represents the variety of the internet is of great importance, popular ranking systems are subjected to manipulations in a way that potentially skews the conclusions made in studies [26].

To prevent such skewness in this study, the Tranco Top Site Ranking system was used as a data source for creating an initial website list containing 35,707 websites. The Tranco Top Site Ranking system has evaluated different popular ranking systems to reduce the fluctuations that occur when composing a ranked list, thereby allowing the research community to work with reliable and reproducible rankings.

This initial list of websites was enriched by the "Google Safe Browsing" (GSB) DB to accurately classify and tag each website. GSB classifies a malicious URL into one of the following five classes: "Malware," "Social engineering," "Unwanted software," "Potentially harmful application," and "Threat type unspecified."

Overall, 697 websites that appeared on the Tranco Top Site Ranking system were classified as "Malware" by GSB. The rest of the websites were not labeled by GSB as risky and were treated as benign websites. Accordingly, the prior probability of a website in this dataset being malicious was 1.95%, as

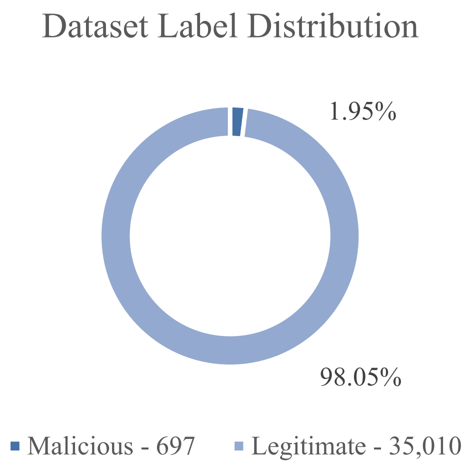


Fig. 2 Dataset label distribution. Only 1.95% of the 35,707 records in the experiment dataset are malicious websites

shown in Fig. 2. While this value properly represents the prior probability for a malicious website in a real-life scenario, this probability produced a significant challenge in the data analysis phase.

3.2 Feature extraction

This paper extended the study of Cohen et al. [24] by developing an algorithm that automatically extracts website features, including full screenshots and image analysis capabilities, in a large-scale operation and enriches each website record with third-party data regarding its operation and metadata. Advanced machine learning (ML) classification models were then applied to determine whether each website was malicious. In particular, the proposed algorithm allowed each website to be accessed to properly extract its design attributes after it has loaded and rendered its content to represent the end-user observation standpoint. Figure 3 emphasizes this extraction.

The algorithm also captured a full screenshot of the website, identified its color scheme and performed image analysis to identify meaningful objects that were being used. In addition, direct enrichment with Alexa services was added to extract traffic-related features for each website.

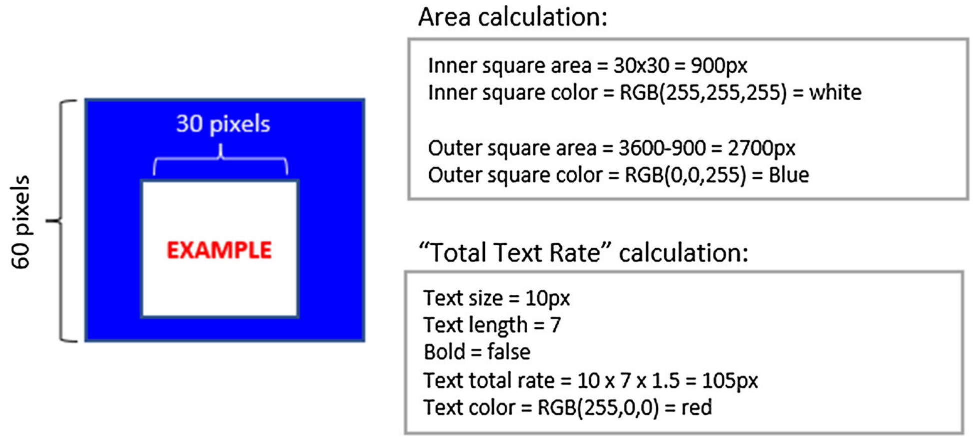
The extracted screenshots were analyzed using the You Only Look Once (YOLO) system [51] together with the Vision API framework. YOLO-V3 is a real-time object detection algorithm consisting of a CNN. This framework was selected due to its ability to provide good results for different types of datasets, the fact that it is far less likely to predict false detection results than other approaches [52] and its proven ability to perform faster than additional leading frameworks, such as Faster region-based CNN (R-CNN) [53–55]. The Vision API was selected based on its ability to represent image contents using structured labels.

The screenshots were analyzed from 2 main perspectives to extract meaningful features: object detection and content classification. Object detection involved identifying meaningful objects in an image, determining their positions and whether they were seen immediately or required scrolling to become visible. Content classification involved the identification of explicit content, such as adult content or violent content, within an image using the Vision API [56, 57]. The output of the above-mentioned feature extraction process was a structured dataset containing the detected images and labels for each website.

On the infrastructure level, the algorithm was enhanced to support such large-scale operations. The proposed algorithmic engine was designed to perform a full scan of one website within a few seconds. It is important to emphasize that built-in waiting times were defined as part of the algorithm to ensure that the website content was loaded and rendered effectively.

Due to the use of parallel computing, the execution of this algorithm for 35,707 websites, including design and schema attribute extraction, screenshot capturing, color distribution determination, and traffic data enrichment, took approximately 23 hours (with a mean time of 14.1 seconds per

Fig. 3 Area and text calculation example



Area calculation:

Inner square area = 30x30 = 900px
Inner square color = RGB(255,255,255) = white

Outer square area = 3600-900 = 2700px
Outer square color = RGB(0,0,255) = Blue

“Total Text Rate” calculation:

Text size = 10px
Text length = 7
Bold = false
Text total rate = 10 x 7 x 1.5 = 105px
Text color = RGB(255,0,0) = red

Table 2 Performance comparison among the neural network models

Model	Dataset characteristics	Minority class prior probability (%)	Accuracy	TPR	FPR
ANN	Imbalanced	1.95	0.98	0.04	0.002
ANN	Balanced	20	0.87	0.77	0.09
ANN	Balanced with under-sampling	70	0.88	0.92	0.23
RNN	Imbalanced with over-sampling	1.95	0.94	0.75	0.06

Table 3 Performance metrics for the bagging classifier and the convergence experiment

	Recall	Specificity	Precision	NPV	FPR	FNR	Accuracy	F1-score
Bagging classifier	0.75	0.98	0.41	0.99	0.02	0.25	0.97	0.53
Convergence experiment	0.83	0.97	0.44	0.99	0.02	0.25	0.97	0.58

website). Following this stage, a complementary image analysis operation was performed offline. This operation used parallel computing as well and took approximately 25 hours (with a mean time of 30 seconds per website).

Overall, the algorithm's output was a structured tabular dataset containing 35,707 website records, where each record consisted of 2900 features. It is important to emphasize that this website dataset consisted of various websites with different geolocations, languages, and web technologies that face different audiences. This variety was essential for capturing the real-life complexity of the World Wide Web. As a result, the built-in variance and the "noise" in this dataset were claimed to be significant.

4 Results

Two different machine learning model types were trained and validated on the collected dataset: an artificial neural network (ANN) and an ensemble classifier consisting of decision trees. Both classifiers were trained using fivefold cross-validation to better utilize the collected data, reduce overfitting, and generalize the model predictions [58].

Due to the efficiency of deep learning models, the suggested approach was tested on an ANN [59] with multiple hidden layers. Neural networks with different hidden layer architectures were trained using 5-fold validation. All networks resulted in high accuracy (above 97.8%) and low FPRs (0.2%) but were only able to correctly identify low rates of malicious websites (2.5–4%). In addition, an analysis of the receiver operating characteristic (ROC) curves yielded by different network architectures indicates that the trained models were affected by overfitting.

The attempt to apply dimensionality reduction and feature selection techniques didn't improve the model's detection capability, nor did it meaningfully contribute in terms of time, memory, or CPU complexity reduction.

This is not surprising, as the effect of class imbalance on the neural network classification performance was previously proven to be detrimental [60, 61]. Two main approaches were previously suggested for handling imbalanced classification problems while using ANN models by substantially increasing the weight of the minority class: supervised oversampling [62] and synthesized data augmentation [63–65]. These approaches share one main disadvantage: active manipulation of the original dataset. This kind of manipulation contradicts the intention of accurately representing a commercial, real-life scenario. In addition, it has been previously suggested that data augmentation techniques do not learn the target distribution [66].

As a proof of concept regarding ANN efficiency in this problem space and to neutralize the imbalanced effect induced without adding synthesized data, more balanced datasets were examined.

The first dataset was a subset of the original dataset consisting of 3485 samples, including all 697 original malicious samples and 2788 legitimate samples that were randomly selected. The prior probability of being a malicious website in this dataset was 10 times greater than that in the previous experiment (20% vs. 1.95%).

To adapt the ANN model to a smaller dataset and prevent overfitting, the feature space and the network architecture were reduced. Accordingly, the trained ANN consisted of 2 hidden layers, while the principal component analysis (PCA) was used for dimensionality reduction, resulting in a feature space consisting of 50 components. As expected, the ANN model performed considerably better on the more balanced dataset and yielded better classification results (recall: 0.819; precision: 0.697; accuracy: 0.891; F1 score 0.753). The balanced model accuracy was inferior to the imbalanced model accuracy, a fact that can be satisfactorily explained by the substantial difference between their minority class prior probabilities.

In addition, we applied an under-sampling technique and trained an ANN model on a second dataset where the Malicious websites were the majority class, with a 70% prior probability of being a malicious website. This second dataset was a subset of the original dataset consisting of 996 samples, including all 697 original malicious samples and 299 legitimate samples that were randomly selected. The ANN with under-sampling performed significantly better in terms of TPR (0.924) and F1 score (0.914) and achieved similar accuracy (0.878) as in the balanced scenario without the under-sampling. However, the FPR increased substantially (0.227). These model settings could be affective in cases where higher FPR are acceptable or in scenarios that involve a layered approach consisting of multiple classifiers.

Another effort to address the crucial effect of class imbalance was made by using A sequential neural network. A recurrent neural network (RNN) was trained with oversampling, and the weight for the minority class of malicious websites was increased. As presented in Table 2, this model was able to detect 75% of malicious websites and achieved higher accuracy (94%) and a lower FPR (0.06) than the balanced ANN model (0.09), as presented in Table 2. This FPR level is similar to those of previously reported methods and is not sufficient for a real-life scenario.

Ensemble models achieve high accuracy by combining a number of base estimators and can increase the reliability of machine learning relative to a single estimator [67].

Bagging (bootstrap aggregation) is a commonly used ensemble classification method that reduces the variance of a decision tree and addresses classification noise [68]. In situations with substantial classification noise, bagging has been found to be superior to boosting and randomization [69]. The algorithm randomly creates several subsets of the given training dataset by sampling with replacement. Every subset is used to train different decision trees, and each different prediction is aggregated into an averaged aggregated prediction. In contrast with the random forest classifier, the bagging classifier does not use a subset of the dataset features and, as a result, can leverage the most significant features for all of its weak classifiers.

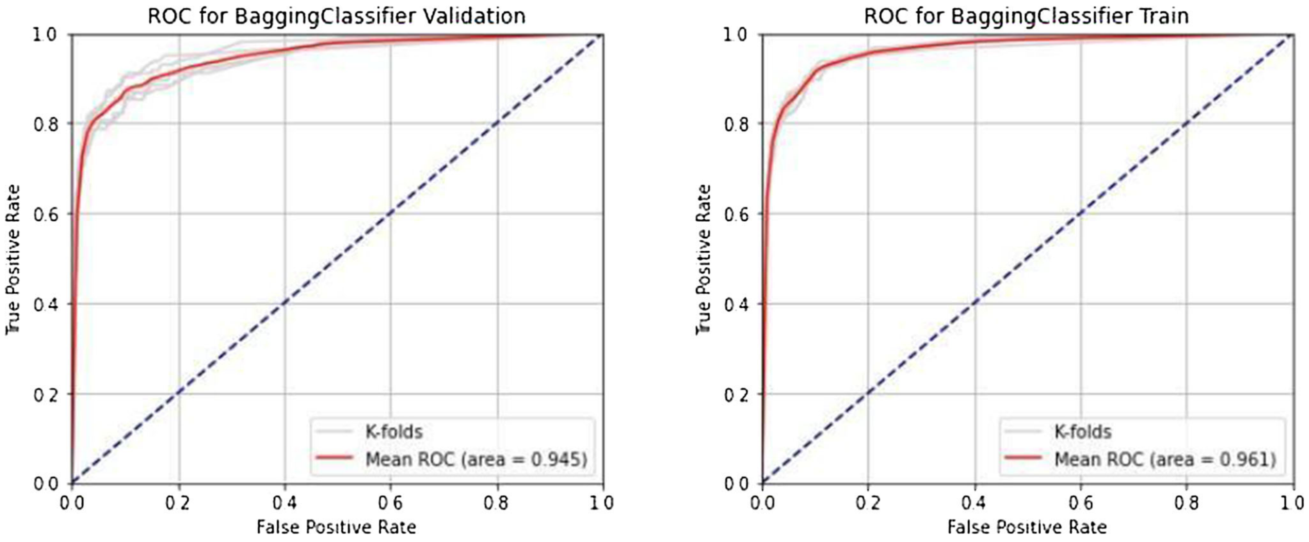


Fig. 4 ROC curves of the bagging classifier for the training and validation folds

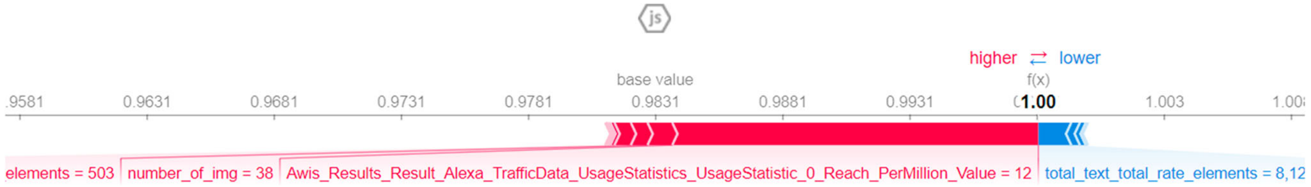


Fig. 5 Example of interactive explainability based on Shapley values. The feature values that increased the predictions are shown in pink, and their visual sizes demonstrate the magnitudes of the corresponding feature effects. The feature values that decreased the predictions are shown in blue. In this example, the model prediction was '1' ('1' represents a malicious website). The biggest impact came from the 3rd-party usage statistics, the high number of images, and the relatively high number of HTML elements compared to the baseline values

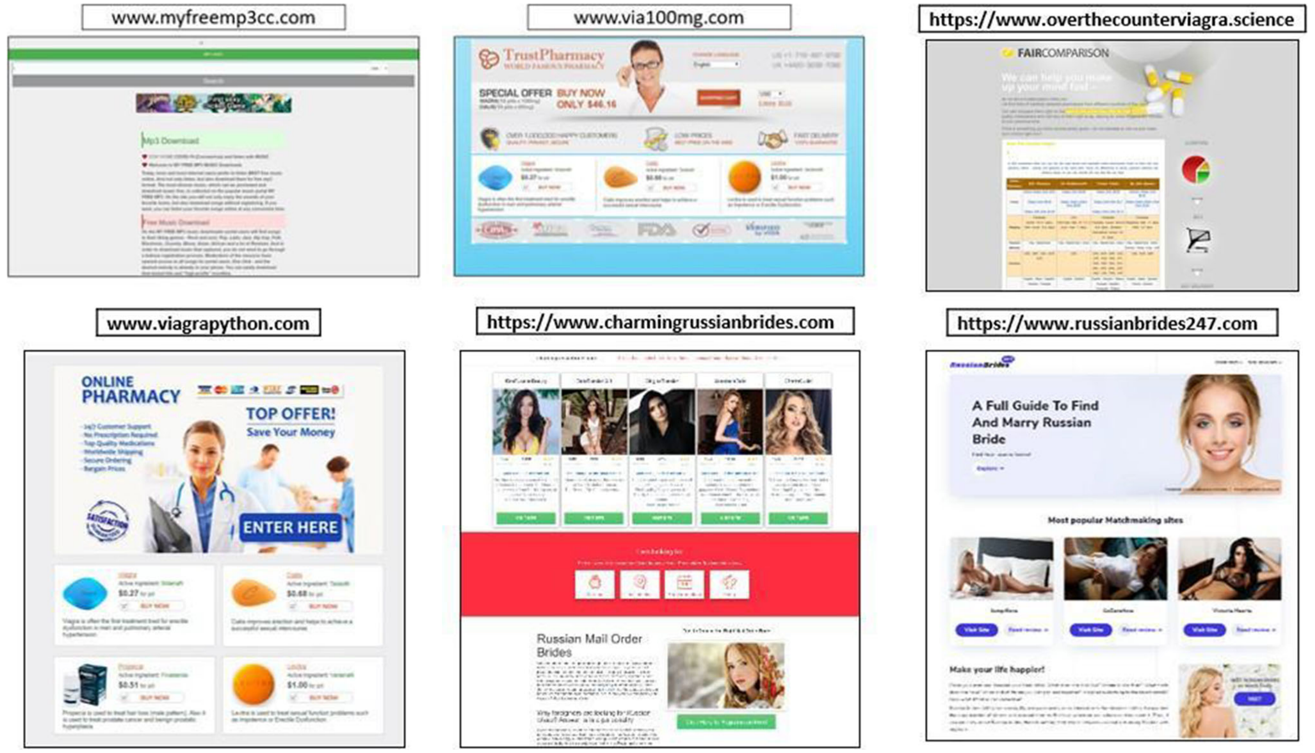

www.myfreemp3cc.com
www.via100mg.com
https://www.overthecounterviagra.science
www.viagrapython.com
https://www.charmingrussianbrides.com
https://www.russianbrides247.com
TrustPharmacy
SPECIAL OFFER BUY NOW ONLY $46.16
FAIRCOMPARISON
ONLINE PHARMACY
TOP OFFER!
Save Your Money
ENTER HERE
Russian Mail Order Brides
A Full Guide To Find And Marry Russian Bride
Most popular Matchmaking sites
Make your life happier!

Fig. 6 Examples of websites that were suggested by the model as malicious. The variety of websites included various content proposals from software downloads via prescription medicine through engagement with human beings. None of the above websites were tagged as malicious by GSB. Note, however, that the fact that the model predicted these websites as malicious is not evidence that they are indeed malicious

The chosen bagging classifier consisted of 10 base estimators. Each estimator was a classification and regression decision tree (CART) with a maximal depth of 4, adjusted to an extremely imbalanced dataset by enforcing a high penalty for classification mistakes produced on the minority class. The CART algorithm was selected due to its advantage in identifying the splitting variables based on searching through all possibilities among the input variables and its ability to leverage its results for explainability purposes. The model was able to successfully detect 75% of all malicious websites while maintaining a low FPR rate of 2% and achieving an overall accuracy of 97.5%, as described in Table 3. Allowing a higher FPR resulted in a higher TPR while maintaining a high accuracy level, as shown in Fig. 4.

To provide explainability for the machine learning model results and to understand the impact of each feature, an implementation of Shapley values [70] for explainable AI was made. Shapley values interpret the impact of having a certain value for a given feature in comparison with the prediction that would have been made if that feature took some baseline value [71]. This implementation also calculates the aggregated contribution in a way that provides insights on a model level.

When analyzing a specific prediction, one can learn what features contributed most and their actual values, as shown in Fig. 5. In this specific example, the model prediction was ‘1’ (‘1’ represents a malicious website). The biggest impact came from the 3rd-party usage statistics, the high number of images and the relatively high number of HTML elements compared to the baseline values.

A deeper dive into the predictions classified as false positives (FPs) revealed that in many cases, malicious websites were not identified by GSB. Out of the predictions considered FPs, 100 websites were randomly selected and manually reviewed. 18% of these websites were identified as malicious, while an additional 21% were identified as suspicious. Among these websites, we found a variety of websites that contained content proposals that might raise suspicion, from software downloads via prescription medicine through engagement with human beings, as demonstrated in Fig. 6.

Accordingly, we conducted a convergence experiment in which the model predictions were reviewed and relabeled. Each prediction classified as an FP was reviewed by going over the explainable AI results and by manually accessing the source code and screenshot of the corresponding website. Then, the allegedly malicious websites were relabeled, and the model was retrained. We learned that the 18% assumption regarding FP predictions that would actually be malicious websites continued to exist through the iterations. However, every iteration discovered additional malicious websites that were classified by GSB as legitimate, and as a result, the experiment did not converge after 5 iterations.

Taking that into consideration, it is reasonable to claim that the model performance was actually higher than that indicated by the above performance metrics. Examining the first iteration of the convergence experiment, as demonstrated in Table 3, the results imply that the model performance will actually be higher in a real-life scenario and that 83% of the malicious websites can be detected by the proposed approach under the same FPR and accuracy measurements.

5 Conclusions

Detecting malicious websites is a challenging and never-ending task. Different techniques and approaches have been suggested to tackle this problem. However, these approaches tend to suffer from a built-in problem: an attacker can use the exact same detection technique to enhance his or her attack vector and evade the defense mechanisms.

The proposed algorithm leverages a fundamental conflict in malicious website operation and widely leverages website design attributes to perform classification.

The suggested approach was validated using a real-life, large-scale and extremely imbalanced dataset. This approach could effectively detect more than 83% of malicious websites while maintaining a low FPR of 2%. In addition, it proved that it could effectively detect malicious and suspicious websites that had previously allegedly slipped under the radar.

The suggested framework offers explainability and can leverage the cybersecurity practitioner's experience and feedback to perform better and respond to emerging threats. In this case, a potential lift of 10% can be achieved relative to a model that does not leverage any end-user feedback.

There are known limitations regarding the use of this approach. First, to implement the proposed approach as part of a business process, dedicated computing resources should be assigned to host and run the algorithm. In addition, future model experiments and enhancements require assembling an additional collection of extracted features from a new set of URLs to accurately represent an up-to-date commercial real-life malicious website identification scenario.

Additional research can be conducted to extend and enhance the suggested framework to provide a classification method of malicious websites into categories, such as phishing websites, financial malware, keyloggers, trojans, and ransomware. The ability to use the developed framework for multiclass website classification can be generalized to address additional business cases outside the cybersecurity domain.

Acknowledgements This paper was partially supported by the Koret Foundation Grant for Digital Living 2030.

Author contributions Naim wrote the main manuscript text and performed conceptualization, investigation, methodology, software, resources, analysis, validation, visualization, and writing. Cohen performed methodology, analysis, validation, and review. Ben Gal performed conceptualization, methodology, validation, writing, administration, and supervision. All authors reviewed the manuscript.

Data availability The dataset generated and analyzed during the current study is available in the Mendeley Data repository [72], https://data.mendeley.com/datasets/thd5mhns6c/1, doi: https://doi.org/10.17632/thd5mhns6c.1.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Xu L, et al.: An evasion and counter-evasion study in malicious websites detection. In 2014 IEEE Conference on Communications and Network Security (2014)
2. Heiderich, M., Frosch, T., Holz, T.: IceShield: detection and mitigation of malicious websites with a frozen DOM. In: Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection (RAID'11), Springer-Verlag, pp 281–300 (2011)
3. Liu, L., et al.: Chrome extensions: threat analysis and countermeasures. In: NDSS (2012)
4. Toreini, E., et al.: DOMtegrity: ensuring web page integrity against malicious browser extensions. Int. J. Inf. Secur. **18**, 801–814 (2019)
5. Dewald A, Holz T, Freiling FC: ADSandbox: sandboxing javascript to fight malicious websites. In: Proceedings of the 2010 ACM Symposium on Applied Computing (2010)
6. Kishore K, Ravi: Browser JS guard: detects and defends against malicious javascript injection based drive by download attacks. In: The Fifth International Conference on the Applications of Digital Information and Web Technologies (2014)
7. Egele, M., Scholte, T., Kirda, E., Kruegel, C.: A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv. **44**(6), 1–42 (2008)
8. Bat-Erdene, M.: Entropy analysis to classify unknown packing algorithms for malware detection. Int. J. Inf. Secur. **16**, 227–248 (2017)
9. Liu, X., Lin, Y., Li, H., Zhang, J.: A novel method for malware detection on ML-based visualization technique. Comput. Secur. **89**, 101682–101682 (2020)
10. Fang, Y., Huang, C., Su, Y., Qiu, Y.: Detecting malicious JavaScript code based on semantic analysis. Comput. Secur. **93**, 101764–101764 (2020)
11. Verma, R., Das, A.: What's in a url: fast feature extraction and malicious url detection. In: Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics (2017)
12. Kim, S., Kim, J., Kang, B.B.: Malicious URL protection based on attackers' habitual behavioral analysis. Comput. Secur. **77**, 790–806 (2018)
13. Chiba, D., Akiyama, M., Yagi, T., Hato, K., Mori, T., Goto, S.: DomainChroma: Building actionable threat intelligence from malicious domain names. Comput. Secur. **77**, 138–161 (2018)

14. Darling, M.: A lexical approach for classifying malicious URLs. In: 2015 International Conference on High Performance Computing & Simulation (HPCS) (2015)
15. Lin, K.Y.: Defense automatic malicious tools based on navigation behavior. In: 24th IEEE International Conference on Advanced Information Networking and Applications, vol. 13, pp. 17–27 (2010)
16. Wen, S., Zhao, Z., Yan, H.: Detecting malicious websites in depth through analyzing topics and webpages. In: Proceedings of the 2nd International Conference on Cryptography, Security and Privacy (2018)
17. Suh-Lee, C., Jo, JY., Kim, Y.: Text mining for security threat detection discovering hidden information in unstructured log messages. In: 2016 IEEE Conference on Communications and Network Security (CNS) (2016)
18. Singh, J., Singh, J.: Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. Inf. Softw. Technol. **121**, 106273 (2020)
19. Mimura, M., Ito, R.: Applying NLP techniques to malware detection in a practical environment. Int. J. Inf. Secur. **21**, 279–291 (2022)
20. Bernardi, M.L., Cimitile, M., Distante, D., Martinelli, F., Mercaldo, F.: Dynamic malware detection and phylogeny analysis using process mining. Int. J. Inf. Secur. **18**(3), 257–284 (2019)
21. Kammerstetter, M., Platzer, C., Wondracek, G.: Vanity, cracks and malware: insights into the anti- copy protection ecosystem. In: Proceedings of the 2012 ACM conference on Computer and communications security (2012)
22. Zhang, X., Wang, Y., Mou, N., Liang, W.: Propagating both trust and distrust with target differentiation for combating link-based web spam. ACM Trans Web **8**(15), 33–33 (2014)
23. Pavlou, P.A., Liang, H., Xue, Y.: Understanding and mitigating uncertainty in online exchange relationships: a principal-agent perspective. MIS Q **31**, 105–136 (2007)
24. Cohen, D., Naim, O., Toch, E., Ben-Gal, I.: Website categorization via design attribute learning. Comput. Secur. **1**(107), 102312 (2021)
25. Invernizzi, L., Comparetti, PM., Benvenuti, S., Kruegel, C., Cova, M., Vigna, G.: Evilseed: a guided approach to finding malicious web pages. In: 2012 IEEE Symposium on Security and Privacy, pp 428– 442 (2012)
26. Pochat, V.L., Goethem, T.V., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: a research- oriented top sites ranking hardened against manipulation. arXiv preprint arXiv:180601156 (2018)
27. Sahoo, D., Liu, C., Hoi, S.C. Malicious URL detection using machine learning: a survey. arXiv preprint arXiv:170107179 (2017)
28. Kim, B.I., Im, C.T., Jung, H.C.: Suspicious malicious web site detection with strength analysis of a javascript obfuscation. Int. J. Adv. Sci. Technol. **26**, 19–32 (2011)
29. Kolbitsch, Clemens, et al.: Rozzle: de-cloaking internet malware. In: 2012 IEEE Symposium on Security and Privacy. IEEE (2012)
30. Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., Ohsita, Y., Murata, M.: Malicious URL sequence detection using event denoising convolutional neural network. 2017 IEEE International Conference on Communications (ICC) (pp. 1–7). IEEE (2017)
31. Moshchuk, A., Bragin, T., Gribble, S.D., Levy, H.M.: A crawler-based study of spyware on the web. In: Proceedings of Network and Distributed System Security Symposium (NDSS) (2006)
32. Wang, Y.M., Beck, D., Jiang, X., Roussev, R.: Automated web patrol with strider honeymonkeys: finding web sites that exploit browser vulnerabilities. In: In NDSS (2006)
33. Eshete, A.B., Villafiorita, K., Weldemariam.: Binspect: Holistic analysis and detection of malicious web pages. In: Security and Privacy in Communication Networks (2013)
34. Ma, J., Saul, K.L., Savage, S., Voelker, G.M.: Learning to detect malicious urls. ACM Trans. Intell. Syst. Technol. **2**(3), 1–24 (2011)
35. Verma, R., Dyer, K.: On the character of phishing URLs: accurate and robust statistical learning classifiers. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, pp. 111–122 (2015)
36. Zhang, Yue, J.I., Hong, L.F., Cranor: Cantina: a content-based approach to detecting phishing web sites. In: Proceedings of the 16th International Conference on World Wide Web (2007)
37. Provos, N., Mavrommatis, P., Rajab, M., Monrose, F.: All your iframes point to us (2008)
38. Guang, X., Jason, O., Carolyn, P.R., Lorrie, C.: CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 1–28 (2011)
39. Whittaker, C., Ryner, B., Nazif, M.: Large-scale automatic classification of phishing pages (2010)
40. Altay, B., Dokeroglu, T., Cosar, A.: Context- sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection. Soft. Comput. **23**(12), 4177–4191 (2019)
41. Saini, A., Gaur, M.S., Laxmi, V., Conti, M.: You click, I steal: analyzing and detecting click hijacking attacks in web pages. Int. J. Inf. Secur. **18**(4), 481–504 (2019)
42. Huang, L.S., Moshchuk, A., Wang, H.J., Schecter, S., JacKson, C.: Clickjacking: attacks and defenses. In: 21st USENIX Security Symposium (USENIX Security 12), pp. 413–428 (2012)
43. Amrutkar, C., Kim, Y.S., Traynor, P.: Detecting mobile malicious webpages in real time. IEEE Trans. Mob. Comput. **16**(8), 2184–2197 (2017)
44. Mcgahagan, J., Bhansali, D., Pinto-Coelho, C., Cukier, M.: A Comprehensive evaluation of webpage content features for detecting malicious websites. In: 2019 9th Latin-American Symposium on Dependable Computing (LADC), pp. 1–10 (2019)
45. Liu, D., Lee, J.H.: CNN based malicious website detection by invalidating multiple web spams. IEEE Access **8**, 97258–97266 (2020)
46. Singh, A.K., and Navneet Goyal.: A comparison of machine learning attributes for detecting malicious websites. In: 2019 11th International Conference on Communication Systems & Networks (COMSNETS). IEEE, (2019)
47. Morales, P.R., Tabourier, L., Ung, S., Prieur, C.: Role of the website structure in the diversity of browsing behaviors. In: Proceedings of the 30th ACM Conference on Hypertext and Social Media, pp. 133–142 (2019)
48. Duckett, J.: HTML & CSS: design and build websites, vol. 15. Wiley Indianapolis, IN (2011)
49. Nixon, R.: Learning PHP, MySQL, JavaScript, and CSS: a step-by-step guide to creating dynamic websites (2012)
50. Chen, M., Ryu, Y.U.: Facilitating effective user navigation through website structure improvement. IEEE Trans. Knowl. Data Eng. **25**(3), 571–588 (2011)
51. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:180402767 (2018)
52. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
53. Nie, Y., Sommella ,P., O'Nils, M., Liguori, C., Lundgren, J.:Automatic detection of melanoma with yolo deep convolutional neural networks. In: 2019 E-Health and Bioengineering Conference (EHB), pp. 1– 4 (2019)
54. Wang, H., Yang, G., Li, E., Tian, Y., Zhao, M., Liang, Z.:High-voltage power transmission tower detection based on faster R-CNN and YOLO-V3. In: Chinese Control Conference (CCC), pp. 8750–8755 (2019a)
55. Wang, Z., Zhang, J., Zhao, Z., Su, F.: Efficient Yolo: a Lightweight Model For Embedded Deep Learning Object Detection. In: 2020

IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1–6 (2020)

56. Chen, S.H., Chen, Y.H.: A content-based image retrieval method based on the google cloud vision api and wordnet. In: Asian Conference On Intelligent Information And Database Systems, Springer, pp. 651– 662 (2017)
57. Bosch, O.J., Revilla, M., Paura, E.: Answering mobile surveys with images: an exploration using a computer vision API. Soc. Sci. Comput. Rev. **37**, 669–683 (2019)
58. Rodriguez, J.D., Perez, A., Lozano, J.A.: Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 569–575 (2009)
59. Hassoun, M.H.: Fundamentals of artificial neural networks. MIT press (1995)
60. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. **106**, 249–259 (2018)
61. Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P.J.: Training deep neural networks on imbalanced data sets. 2016 International Joint Conference on Neural Networks (IJCNN) pp. 4368–4374 (2016)
62. Nguyen, G.H., Bouzerdoum, A., Phung, S.L.: A supervised learning approach for imbalanced data sets. 19th International Conference on Pattern Recognition, pp. 1–4 (2008)
63. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C. Bagan: data augmentation with balancing gan. arXiv preprint arXiv:180309655 (2018)
64. Ali-Gombe, A., Elyan, E.: MFC-GAN: class- imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing **361**, 212–221 (2019)
65. Wang, J., Li, S., Han, B., An, Z., Bao, H., Ji, S.: Generalization of deep neural networks for imbalanced fault classification of machinery using generative adversarial networks. IEEE Access **7**, 111168–111180 (2019)
66. Arora, S., Zhang, Y.: Do gans actually learn the distribution? an empirical study. arXiv preprint arXiv:170608224 (2017)
67. Taherkhani, A., Cosma, G., Mcginnity, T.M.: AdaBoost-CNN: an adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. Neuro-computing **404**, 351–366 (2020)
68. Abellán, J., Masegosa, A. R.: Bagging decision trees on data sets with classification noise. In International Symposium on Foundations of Information and Knowledge Systems (pp. 248–265). Springer, Berlin, Heidelberg (2010)
69. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach. Learn. **40**, 139–157 (2000)
70. Hart, S.: Shapley value. In Game theory (pp. 210–216). Palgrave Macmillan, London (1989)
71. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: From local explanations to global understanding with explainable AI for trees. Nat. Mach. Intel. **2**(1), 56–67 (2020)
72. Naim, O.r.: Malicious website identification using design attribute learning. In: Mendeley Data, vol. V1, doi: https://doi.org/10.17632/thd5mhns6c.1(2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.