http://doi.org/10.35784/iapgos.8143

received: 24.07.2025 | revised: 22.08.2025 | accepted: 16.09.2025 | available online: 30.09.2025

THE EFFECTIVENESS OF MACHINE LEARNING IN DETECTING PHISHING WEBSITES

Jacek Łukasz Wilk-Jakubowski[1], Aleksandra Sikora[2], Dawid Maciejski[3]

[1]Kielce University of Technology, Department of Information Systems, Kielce, Poland, [2]Kielce University of Technology, Department of Computer Science, Electronics and Electrical Engineering, Kielce, Poland, [3]Kielce University of Technology, Faculty of Electrical Engineering, Automatic Control and Computer Science, Kielce, Poland

Abstract. *Phishing poses a significant risk in the field of digital security, requiring effective methods for identifying fraudulent websites. This study evaluated the performance of nine machine learning classification models in the context of phishing website detection. Two different input datasets were prepared: the first included the full HTML code, while the second was based on a set of features extracted from that code. The analysis revealed that models trained on the extracted features achieved nearly twice the detection performance compared to those operating on raw HTML code. The use of majority voting further improved classification effectiveness. The study results confirm that proper feature selection and the integration of outputs from multiple models significantly enhance the effectiveness of systems for detecting online threats.*

Keywords: phishing, machine learning, website classification, feature analysis, network security, threat detection

SKUTECZNOŚĆ UCZENIA MASZYNOWEGO W WYKRYWANIU STRON PHISHINGOWYCH

Streszczenie. *Zjawisko phishingu stanowi poważne ryzyko w dziedzinie bezpieczeństwa cyfrowego, co wymaga efektywnych sposobów identyfikacji fałszywych witryn. W ramach badania oceniono efekty dziewięciu modeli klasyfikacyjnych opartych na uczeniu maszynowym w kontekście rozpoznawania stron phishingowych. Przygotowano dwa różne zestawy danych wejściowych: pierwszy obejmował pełny kod HTML, natomiast drugi opierał się na zestawie cech wydobytych z tego kodu. Przeprowadzona analiza ujawniła, że modele trenowane na wyodrębnionych cechach osiągają niemal dwukrotnie lepsze wyniki w wykrywaniu od modeli działających na surowym kodzie HTML. Wykorzystanie głosowania większościowego przyczyniło się do dalszej poprawy skuteczności klasyfikacji. Rezultaty badań potwierdzają, że odpowiedni dobór cech oraz integracja wyników z wielu modeli znacząco podnoszą efektywność systemów identyfikujących zagrożenia w Internecie.*

Słowa kluczowe: phishing, uczenie maszynowe, klasyfikacja stron internetowych, analiza cech, bezpieczeństwo sieci, detekcja zagrożeń

Introduction

In reference to previous studies on the architecture of broadband information systems used, among other things, in the context of crisis management [3], the importance of reliable communication and resilient IT (Information Technology) infrastructure in difficult situations was emphasized. When examining the broader context of information systems engineering, it becomes clear that effective threat detection in computer networks – such as phishing websites – depends not only on classification methods but also on the stability of the transmission infrastructure. These issues were also addressed in an earlier article on VSAT (Very Small Aperture Terminal) satellite networks and data transmission problems under limited connectivity conditions [14, 20].

In such circumstances, the security of transmitted data becomes extremely important, and the effective detection of digital threats such as phishing plays a key role in protecting systems and users. This publication expands on the topic, focusing on the application of machine learning techniques to detect fake websites, which may be part of broader attacks – even during crises or disruptions.

The dynamic development of Internet services and IoT (Internet of Things) devices creates new opportunities for cybercriminals. One of the most common threats is phishing, which involves impersonating legitimate websites to, for example, steal confidential user data or disrupt system operations [6].

A phishing website is one that impersonates a legitimate online service – such as a bank, store, or social networking site – to deceive users and obtain their data, such as logins, passwords, or credit card numbers. In contrast, a legitimate website is an authentic, trusted service whose purpose is to provide functions in accordance with its stated mission and to maintain the security of user data [5, 8].

This article evaluates the effectiveness of a developed system for detecting phishing websites using selected machine learning models. The system was based on the analysis of HTML (HyperText Markup Language) code from specific URLs (Uniform Resource Locators). Two different approaches to representing input data were used. The first focused on analyzing the textual content of the websites, converted into numerical representation using the TF-IDF (Term Frequency-Inverse Document Frequency) technique, which accounts for both the frequency of word occurrence and its importance in the context of the entire dataset. The second approach involved manually selecting a set of features from the HTML code, such as the number of forms, embedded elements, text length, and the presence of suspicious keywords.

The project applied nine popular classification models: Support Vector Machine (SVM) [15], XGBoost (Xtreme Gradient Boosting) [11], Random Forest [4], Logistic Regression [7], AdaBoost [15], Gradient Boosting [7], LightGBM (Light Gradient Boosting Machine) [4], CatBoost (Categorical Boosting) [11], and MLP (MultiLayer Perceptron) [4]. Among these models, Random Forest has repeatedly demonstrated very high classification accuracy in the literature [13], achieving over 99% accuracy in many tests. The models were trained on a properly prepared dataset based on selected columns from a dataset available on the Kaggle platform.

The analysis of the results allowed for a comparison of both data representation approaches and an assessment of their effectiveness. The results showed that the approach based on features extracted from HTML code achieved higher accuracy in identifying threats and better capability in distinguishing legitimate websites. Therefore, the system based on this method proves to be more resistant to techniques used by cybercriminals and constitutes an effective tool in the fight against phishing.

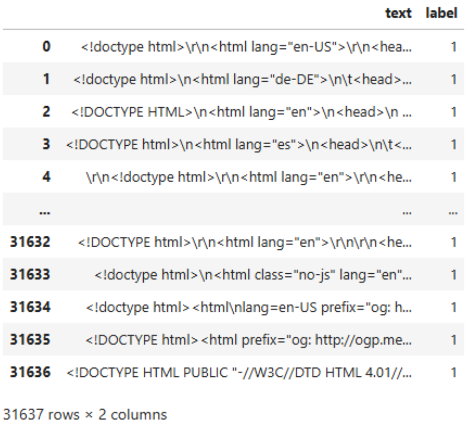
1. Data Preprocessing and Machine Learning Model Development

1.1. Data Processing and HTML Code Acquisition

To train the system for phishing website identification, the experiments used a dataset available in the Kaggle repository – see [2]. The dataset was last updated in March 2024 and contains over 100,000 URL records classified as phishing or legitimate. The data was collected from various online sources, such as blacklists and public website directories. The dataset includes URLs labeled as malicious (with the label column marked as 1) or safe (with the label column marked as 0).

During the initial analysis, two key columns were selected from the dataset [2]: URL and label. Based on these, a script was prepared to retrieve the HTML content of each listed website.

The retrieval process involved sending HTTP requests to the specified URLs. The server responses containing the full HTML code were saved in a new column named text [10]. The label column was retained as the target variable used for training and testing the classifiers.



	text	label
0	<!doctype html>\r\n<html lang="en-US">\r\n<hea...	1
1	<!doctype html>\n<html lang="de-DE">\n\t<head>...	1
2	<!DOCTYPE HTML>\n<html lang="en">\n<head>\n ...	1
3	<!DOCTYPE html>\n<html lang="es">\n<head>\n\t<...	1
4	\r\n<!doctype html>\r\n<html lang="en">\r\n<he...	1
...	...	...
31632	<!DOCTYPE html>\r\n<html lang="en">\r\n\r\n<he...	1
31633	<!doctype html>\n<html class="no-js" lang="en"...	1
31634	<!doctype html> <html\nlang=en-US prefix="og: h...	1
31635	<!DOCTYPE html> <html prefix="og: http://ogp.me...	1
31636	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//...	1

31637 rows × 2 columns

Fig. 1. Sample fragment of the selected dataset

As a result of processing the input data, a new dataset was created, presented in Fig. 1, which contains all the necessary information for the subsequent stages of analysis and modeling. This dataset served as the basis for training and comparing the performance of selected machine learning models.

1.2. Methods of input data processing

To effectively detect phishing websites, two independent approaches to input data representation were developed, both based on the analysis of HTML code. Each method focused on a different layer of information contained in the HTML document [18]. This approach aligns with the findings of previous studies, which showed that the effectiveness of phishing classifiers strongly depends on the method of data representation – both structural and content-based [1, 13].

1.3. Text extraction from HTML code

The first approach involved extracting the plain text content from the HTML code, omitting tags, styles, and scripts. This content, obtained from the Document Object Model (DOM), included information visible directly to the user visiting the website. The extracted text was then transformed into a numerical representation using the TF-IDF method. This technique considers not only the frequency of individual words within a document but also their uniqueness relative to the entire dataset. As a result, the model was able to identify characteristic words or phrases frequently found on phishing websites, while ignoring common and less significant terms. The resulting text vectors were used as input data for training the classification models. However, it should be noted that approaches based solely on text analysis may be less resilient to obfuscation or random manipulative techniques, as demonstrated, for example, in the work by Tashtoush et al., where the authors applied statistical n-gram analysis instead of traditional text representation [18].

1.4. Extraction of structural features from HTML

The second approach focused on examining the technical elements of the HTML code structure. Instead of analyzing textual content, a set of syntactic and semantic features was extracted, which may suggest the presence of specific patterns commonly found on phishing websites. These features relate to the organization and structure of the website's source code, regardless of its visual appearance. Such properties are often overlooked in traditional content analysis, yet they can provide valuable insights into unusual or suspicious mechanisms behind the functioning of a given webpage.

The feature set included, among others:

- the number of HTML forms – phishing websites often use forms to collect login credentials from users. A high number of <form> tags may suggest an attempt to manipulate user interaction, for example, through hidden or deceptive login forms,
- the number of nested iframes (<iframe>) – nested iframes can be used to load content from various websites, often without the user's knowledge. In phishing attacks, they may serve to conceal malicious elements or create deceptive interfaces that mimic legitimate websites,
- the total length of the page code and its individual sections (e.g., <head>, <body>) – an unusually short or excessively long website code, as well as inconsistencies between different sections of the document, may indicate automatically generated content or attempts to conceal malicious code. Analyzing the length of individual elements enables the detection of irregularities in the site's structure,
- the number of references to external sources (e.g., links to scripts or styles from domains other than the main site) – phishing websites often use external sources to reduce resource consumption or to imitate the appearance of well-known brands. A high number of such references may indicate an attempt to impersonate another site or to use content outside the control of the website owner,
- the ratio of visible text length to the total code size – if a website contains very little visible text compared to the overall size of its HTML code, it may indicate the presence of hidden information or excessive coding. Such practices are often used in phishing attacks to conceal malicious activities or to deceive search engine results,
- the presence of tags frequently exploited in phishing, such as onmouseover, window.open(), and javascript – interactive elements, such as onmouseover (triggered when the cursor hovers over an element), window.open() (opening new browser windows), or embedded JavaScript scripts, are often used to redirect Internet users or to display false information. Their excessive use can be characteristic of phishing scams.

This approach is supported in the literature, where structural features – such as the presence of frames, forms, or scripts – are considered effective indicators of potentially malicious content [1, 13].

2. Experimental setup

To evaluate the effectiveness of systems identifying safe content, a dataset of legitimate and well-known websites was used, including Google, Wikipedia, YouTube, and popular news portals, as well as a dataset available on the Kaggle platform – see reference [16]. These examples formed a representative sample of websites considered non-threatening, aiming to assess the classifiers' ability to distinguish content that does not pose a phishing threat during everyday Internet use. Testing on websites deemed safe allowed for thorough verification of whether the models produce false alarms, which is crucial in real-world applications. Excessive warnings about safe sites could lead to reduced user trust in the system.

In parallel, the classifiers' performance was analyzed on a dataset containing phishing websites, sourced from public repositories such as PhishTank [12]. This enabled the assessment of how well the models detect malicious content under conditions resembling real-world scenarios. Particular emphasis was placed on detection effectiveness and minimizing missed threats.

It is worth noting that this study utilized a preconfigured system in which the models were previously trained on a specific dataset [2]. Then, using the saved and previously trained classifiers, the system analyzed new, unknown websites. This approach simulates a real-world scenario of a security system operating in predictive mode – assessing potential threats based on previously acquired knowledge. This allowed for a realistic evaluation of the models' effectiveness in the context of practical applications.

3. Evaluation of phishing website detection effectiveness

The conducted studies found that models trained on HTML code features have a significant advantage, identifying 831 phishing sites (see Fig. 2), while text-analysis-based models detected only 436 cases (see Fig. 3). Importantly, despite their higher effectiveness, feature-based models also maintained a satisfactory level of precision, misclassifying only 316 out of 4,173 legitimate sites as phishing (see Fig. 4).

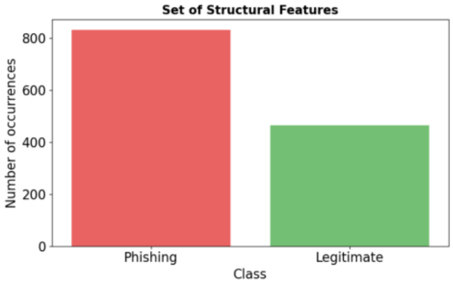


Fig. 2. Majority voting chart for phishing websites using structural features

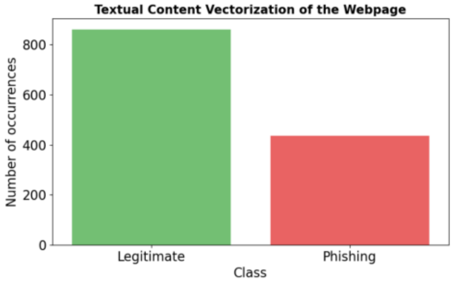


Fig. 3. Majority voting chart for phishing websites using text vectorization

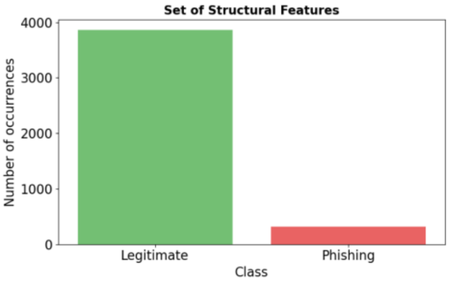


Fig. 4. Majority voting chart for legitimate websites using HTML features

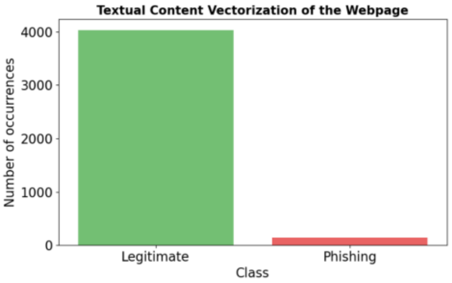


Fig. 5. Majority voting chart for legitimate websites using text vectorization

The versions of models that utilized text extracted from HTML code achieved very high effectiveness in classifying legitimate websites. In the case of the version using text-based representation, the vast majority of addresses were correctly classified as legitimate – as many as 4,028 cases, while only 145 addresses were incorrectly labeled as phishing (see Fig. 5). Overall, 96.5% of legitimate sites were correctly identified, and only 3.5% were mistakenly classified as phishing (see Fig. 6). However, these models struggled to identify actual threats. For URLs associated with phishing, only 33.6% were successfully detected as malicious, while as much as 66.4% of phishing sites were incorrectly classified as safe (see Fig. 7). Such low detection accuracy increases the likelihood that a real phishing attack will go unnoticed by the system and reach the end user without any warning.

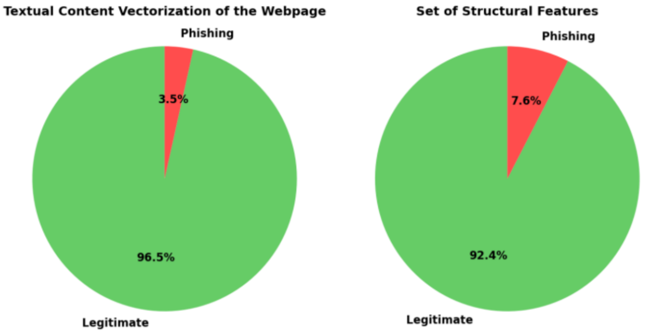


Fig. 6. Percentage results of majority voting for legitimate websites

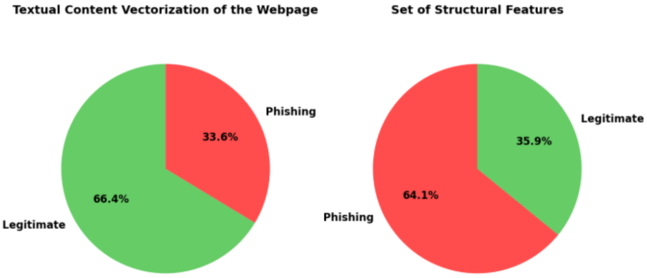


Fig. 7. Percentage results of majority voting for phishing websites

The differences in the effectiveness of both analytical methods stem from their distinct theoretical foundations. The TF-IDF-based approach focuses on analyzing the textual content of a website, namely the words and phrases present in the HTML code. Such analysis enables the detection of common expressions or suspicious language patterns that may indicate an attempted fraud. However, this method is vulnerable to certain manipulations, particularly in the case of modern phishing websites, which often imitate the appearance and content of legitimate pages. In such cases, the text may be nearly identical to the original, which significantly hampers the accurate detection of threats based solely on content analysis.

An alternative solution is structural analysis, which relies on a defined set of technical features, such as the number of forms, links, iframe elements, text length, and the presence of suspicious tags and keywords. By focusing on the structure of the HTML and its technical components, this approach becomes more difficult for cybercriminals to bypass. Structural analysis makes it possible to identify unusual patterns that are characteristic of phishing websites, thereby increasing the likelihood of successfully detecting a threat even when the page visually resembles the original. However, it is important to emphasize that static HTML code analysis has its limitations. It does not allow for the detection of attacks that use dynamic mechanisms such as JavaScript scripts, dynamically loaded DOM content, or other client-side techniques.

4. Analysis of the effectiveness of individual classification models

The charts illustrate the classification results for legitimate websites (see Fig. 8 and 9), showing that most models correctly classify the samples. For statistical feature sets extracted from HTML, the accuracy is slightly lower but still remains high, with most models effectively identifying legitimate sites.

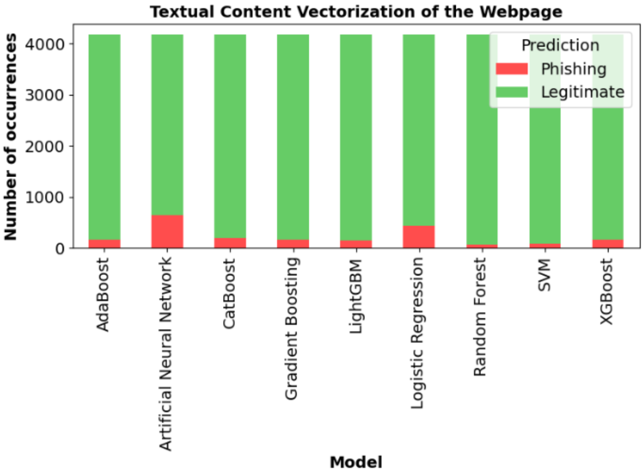


Fig. 8. Comparison of models based on results for legitimate websites using text vectorization

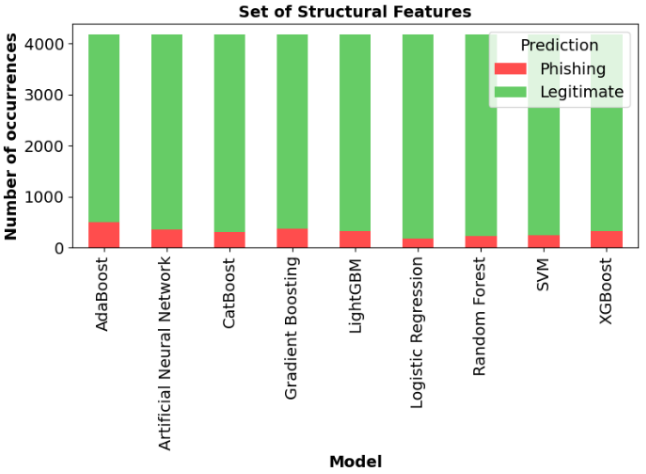


Fig. 9. Comparison of models based on results for legitimate websites using HTML features

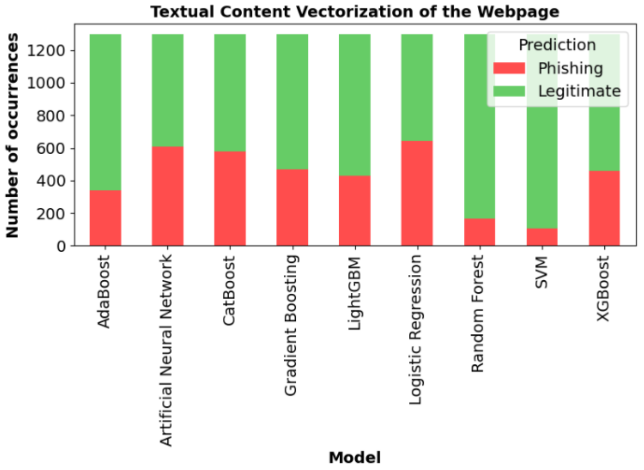


Fig. 10. Comparison of models based on results for phishing websites using text vectorization

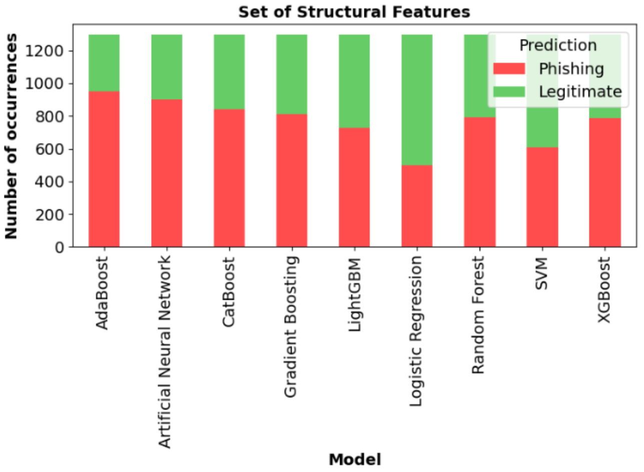


Fig. 11. Comparison of models based on results for phishing websites using HTML features

In contrast, regarding phishing websites, models based on HTML (see Fig. 10) text vectorization demonstrate low effectiveness, with the majority misclassifying malicious samples. Particularly poor performance is observed for models such as Random Forest and SVM.

On the other hand, when using statistical features extracted from HTML (see Fig. 11), the classification accuracy for phishing sites is significantly better, although some models, such as Logistic Regression and SVM, also exhibit noticeably lower precision.

5. Conclusions

The results of the conducted study suggest that, under the examined conditions, classification based on selected structural elements of HTML code may prove to be more effective compared to the TF-IDF-based content analysis technique in identifying phishing websites. Models trained using HTML features detected 64.1% of malicious sites, whereas models based on text vectorization achieved an effectiveness of 33.6%. It should be noted, however, that these results apply to a specific testing environment and a selected dataset. Moreover, the study did not account for dynamic analysis, such as JavaScript execution, and was limited to static HTML code, which may also influence the final assessment of the effectiveness of each approach.

Despite higher sensitivity, models based on HTML structural analysis still achieved a satisfactory level of accuracy – only 7.6% of authentic websites were incorrectly classified as phishing. On the other hand, models using TF-IDF better identified legitimate sites; however, in the context of security, effectiveness in threat detection is key, even if it means an increase in false alarms.

The approach focusing on the analysis of structural features of HTML code shows potential for practical use, with models such as Support Vector Machine, Random Forest, LightGBM, and CatBoost proving particularly effective.

It is worth noting that combining various approaches – such as behavioral analysis, heuristics, or content analysis – could further enhance detection effectiveness and reduce the number of false positives.

It should be emphasized that similar results were observed in studies [19], which analyzed the development of information systems in terms of their architecture and data transmission, indirectly highlighting the importance of structural features in assessing information-related threats.

The achieved results may serve as a starting point for the development of resilient information systems, especially in crisis situations and during various disruptions. Referring to previous analyses concerning the architecture of broadband communication systems used in crisis management [3, 17], it is worth emphasizing that the integration of effective phishing detection techniques could enhance the security level of end users, both in mobile and satellite systems as well as in isolated areas. Implementing such solutions in VSAT systems might provide an additional layer of protection in combating phishing attacks targeting critical infrastructure.

References

[1] Albishri A. A., Dessouky M. M.: A comparative analysis of machine learning techniques for URL phishing detection. Engineering, Technology & Applied Science Research 14(6), 2024, 18495–18501 [https://doi.org/10.48084/etasr.8920].

[2] Arvind N.: Phiusiil phishing URL dataset (Version 1) [https://www.kaggle.com/datasets/ndarvind/phiusiil-phishing-url-dataset] (available: 19.03.2025).

[3] Carreras-Coch A., et al.: Communication Technologies in Emergency Situations. Electronics 11(7), 2022, 1155 [https://doi.org/10.3390/electronics11071155].

[4] Géron A.: Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow. Wyd. 3, Helion, 2023.

[5] Hadnagy C., Fincher M., Dreeke R.: Mroczne odmęty phishingu. Nie daj się złowić! Helion, 2018.

[6] Hadnagy C.: Social engineering: the art of human hacking. Wiley Publishing, 2011.

[7] Hastie T., Tibshirani R., Friedman J.: The elements of statistical learning: data mining, inference, and prediction. Springer, 2009.

[8] Jha A.: Fight fraud with machine learning. Manning Publications, 2023.

[9] Liu B.: Web data mining: exploring hyperlinks, contents, and usage data. 2nd ed., Springer, 2011.

[10] Mitchell R.: Web scraping with Python: collecting data from the modern web. 2nd ed., O’Reilly Media, 2018.

[11] Murphy K.P.: Machine learning: a probabilistic perspective. MIT Press, 2012.

[12] PhishTank: Phishing website database [https://phishtank.org/] (accessed: 22.03.2025).

[13] Reyes-Dorta N., Caballero-Gil P., Rosa-Remedios C.: Detection of malicious URLs using machine learning. Wireless Networks 30, 2024, 7543–7560 [https://doi.org/10.1007/s11276-024-03700-w].

[14] Sasanuma M., et al.: Research and development of very small aperture terminals (VSAT) that can be installed by easy operation during disasters – Issues and the solutions for implementing simple and easy installation of VSAT earth station. IEICE 112(440), 2013, 1–3.

[15] Shalev-Shwartz S., Ben-David S.: Understanding machine learning: from theory to algorithms. Cambridge University Press, 2014.

[16] Shashwat A.: Web page phishing detection dataset (Version 1) [https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset] (available: 23.03.2025).

[17] Suematsu N., et al.: Multi-mode SDR VSAT against big disasters. European Microwave Conference’13, Nuremberg, 2013 [https://doi.org/10.23919/EuMC.2013.6686788].

[18] Tashtoush Y., et al.: Exploring low-level statistical features of n-grams in phishing URLs: a comparative analysis with high-level features. Cluster Computing 27, 2024, 13717–13736 [https://doi.org/10.1007/s10586-024-04655-5].

[19] Wilk-Jakubowski J. Ł.: A review on information systems engineering using VSAT networks and their development directions. Yugoslav Journal of Operations Research 31(3), 2021, 409–428 [https://doi.org/10.2298/YJOR200215015W].

[20] Wilk-Jakubowski J. Ł.: Broadband satellite data networks in the context of the available protocols and digital platforms. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska – IAPGOŚ 2, 2021, 56–60 [https://doi.org/10.35784/iapgos.2630].

D.Sc. Jacek Łukasz Wilk-Jakubowski
e-mail: jwilk@tu.kielce.pl

He is an associate professor at the Kielce University of Technology, Faculty of Electrical Engineering, Automatic Control and Computer Science, Department of Information Systems. He was awarded the Doctor of Technical Science degree (with the specialization in ICT, Teleinformatics, Data Transmission and Signal Processing) and doctor of science (habilitation) degree in the Informatics and Computer Science discipline. He is the author of several inventions that have been granted protection by the Patent Office, participant of many national and international conferences and projects, and laureate of several awards, among others for patents. He is the author more than 90 scientific publications (including 5 monographs, 5 chapters in monographs, as well as more than 80 papers).

https://orcid.org/0000-0003-1275-948X

Ph.D. Aleksandra Sikora
e-mail: asikora@tu.kielce.pl

She is an assistant professor at the Kielce University of Technology in the Faculty of Electrical Engineering, Automatic Control and Computer Science, Department of Computer Science, Electronics and Electrical Engineering. She has participated in numerous IT projects integrating scientific research with student education in areas such as cybersecurity, big data, machine learning, and digital metrology.

https://orcid.org/0000-0003-0145-527X

M.Sc. Dawid Maciejski
e-mail: dawidmaciejski0@proton.me

A graduate of second-cycle studies in Computer Science, specializing in Cybersecurity, at the Faculty of Electrical Engineering, Automation, and Computer Science at the Kielce University of Technology, completed in 2025. His main areas of interest include developing machine learning methods, issues related to computer networks, and user security on the Internet.

https://orcid.org/0009-0003-6426-2552