Detecting phishing URLs with self-organizing models*

Serhii Buchyk[1,*,†], Serhii Toliupa[1,†], Anastasiia Shabanova[1,†], and Oleksandr Buchyk[1,†]

[1] *Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., 01033 Kyiv, Ukraine*

Abstract

Phishing remains one of the most pervasive and rapidly evolving cyber threats, leveraging deceptive URLs to manipulate users into divulging sensitive information. This paper introduces a novel, hybrid detection framework that constructs a structured "portrait" of each URL by extracting 12 discriminative features-spanning structural, semantic, technical, and morphological dimensions-including protocol type, presence of risk-related keywords (e.g., login, verify), top-level domain reputation, URL length, subdomain count, and domain age obtained via WHOIS. The framework integrates three complementary, inherently interpretable models: Logistic Regression (LR) for transparent, weight-based decision reasoning; the Group Method of Data Handling (GMDH) to capture complex nonlinear feature interactions in uncertain classification cases (e.g., probabilities near 0.5); and Self-Organizing Maps (SOM) for unsupervised clustering and visual risk landscape analysis. Evaluated on a balanced dataset of 1,000 URLs (500 phishing, 500 legitimate), the system achieves high performance-GMDH yields 98.80% accuracy and 97.36 F1-score, while LR provides 98.73% accuracy with full explainability. The SOM component reveals "hot spots" of phishing concentration, enabling intuitive risk visualization and policy-driven response mechanisms. By harmonizing accuracy, adaptability, and interpretability, this approach addresses critical gaps in existing black-box AI systems and aligns with the stringent transparency requirements of security-critical domains such as banking, e-government, and enterprise cybersecurity. The modular design also supports real-time deployment, offline analysis, and dynamic model refinement through user feedback.

Keywords

phishing URLs, feature portrait, self-organization, logistic regression, GMDH, SOM, explainable AI.

1. Introduction

Phishing is one of the most widespread and fast – evolving cyber threats. According to ENISA's Threat Landscape Report (2023) [1], phishing accounted for more than 36% of reported cybersecurity incidents in Europe, while CERT – UA [2] recorded that in Ukraine alone phishing was responsible for over 40% of user – targeted attacks in 2022. Studies also indicate that the average lifespan of a phishing domain rarely exceeds 12-24 hours, which makes reactive defenses ineffective.

A representative case observed in 2023 further illustrates this problem: attackers targeted Ukrainian banking users by distributing links via SMS messages that led to fake web portals visually identical to popular online banking services [3]. Victims who entered their credentials unknowingly transferred them to attackers, resulting in direct financial losses. Such incidents demonstrate the sophistication of modern phishing campaigns, which combine technical tricks with social engineering.

To address these challenges, in our previous work, we proposed a mathematical approach for assessing the suspiciousness of phishing links and explored its practical application in detection tasks. It has been both theoretically and practically proven that any textual information can be represented as an array, a vector, or a set of elements. These methods remain effective across both initial and derivative code representations, as they are universal due to their flexible algorithmic adaptability and the metamathematical foundation of laws and axioms. At the same time, one should rely on the direct advantages and contextual requirements of their application, since their use may be irrelevant at certain critical points. As a last resort, the decision should be supported by

* *CSDP'2026: Cyber Security and Data Protection, May 7, 2026, Lviv, Ukraine*

* Corresponding author.

† These authors contributed equally.

buchyk@knu.ua (S. Buchyk); tolupa@i.ua (S.Toliupa); shabanovaa@knu.ua (A. Shabanova); alex8sbu@knu.ua (O. Buchyk)

0000-0003-0892-3494 (S. Buchyk); 0000-0002-1919-9174 (S. Toliupa); 0009-0008-4962-569X (A. Shabanova); 0000-0001-7102-2176 (O. Buchyk)

efficiency proportionality coefficients or by categorizing the analyzed information and correlating it with the most suitable method for the given case [4].

Nevertheless, traditional countermeasures, such as blacklists and heuristic filters, are still widely used in practice. While they provide partial protection, both approaches suffer from inherent limitations [5]. Blacklists are effective only against previously identified threats and cannot respond to the constant emergence of new, short – lived domains. Heuristic – based methods, although fast, rely on predefined rules that are easily bypassed by adversaries. Therefore, phishing URL detection should be considered not as an isolated filtering mechanism, but as one component of a broader defense-in-depth architecture, where AI-supported modules can assist in identifying and prioritizing cyber threats [6]. The typical lifecycle of a phishing domain – including creation, exploitation, blocking, and subsequent rotation – is illustrated in Figure 1.

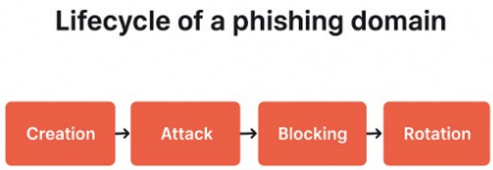

Figure 1: Lifecycle of a phishing domain: from creation and attack to blocking and subsequent rotation.

Similar AI-driven approaches are also actively applied in web application security, where automated models support vulnerability detection, attack classification, and risk-oriented decision-making [7]. In recent years, machine learning (ML) has been increasingly applied to phishing detection, offering improved adaptability and accuracy [8]. However, many of the high – performing models – particularly deep learning architectures – are treated as "black boxes." Their lack of interpretability poses a barrier to practical adoption in critical infrastructures, where transparency of decision – making is required. Recent studies on practical cyberattack detection environments also confirm that security-oriented detection systems require not only high accuracy, but also traceable analytical workflows and reproducible testing conditions [9].

To address these challenges, this study introduces a hybrid detection framework that combines the advantages of structured feature modeling and self – organizing learning principles. By representing each URL as a structured portrait of features and applying a combination of Logistic Regression (LR), the Group Method of Data Handling (GMDH), and Self – Organizing Maps (SOM), the framework aims to balance accuracy, adaptability, and interpretability.

2. State of the Art in Phishing Detection

Over the past two decades, phishing detection has evolved through several methodological stages. The earliest solutions relied on blacklists of malicious domains and URLs, which provided a simple yet effective defense against already known threats. However, these approaches are inherently reactive: they cannot cope with the dynamic nature of phishing, where malicious domains often have a lifespan of only a few hours.

To overcome this limitation, heuristic rule – based methods were introduced. Such systems analyze the structural and lexical properties of URLs, for example the length of the string, the number of subdomains, or the presence of suspicious keywords (e.g., login, verify). While these rules can detect novel phishing attempts, they are static and therefore prone to evasion once adversaries adjust their strategies.

In recent years, machine learning (ML) – based approaches have become the dominant direction. A wide range of algorithms has been applied, from classical models such as Decision Trees, Random Forests, and Logistic Regression, to more complex architectures including Support Vector Machines and Deep Neural Networks. ML methods demonstrate high accuracy and adaptability, but they often suffer from limited interpretability. This is particularly problematic in

critical domains such as banking, e-government, and enterprise cybersecurity, where transparent decision – making is essential. Similar transparency and reliability requirements are also relevant for guaranteed information systems used in distance learning and other digital public-service environments [10].

To address these shortcomings, researchers have increasingly turned to explainable AI (XAI) techniques [11]. At the same time, the growing use of large language models and other complex AI components in cybersecurity raises additional concerns regarding robustness, automated testing, and vulnerability assessment of AI-based systems [12]. For phishing detection, interpretable models such as Logistic Regression (LR), polynomial networks (e.g., GMDH), and clustering methods like Self – Organizing Maps (SOM) are of particular interest. They not only achieve competitive accuracy but also provide insight into which URL features contribute most to classification.

Table 1 provides a comparative overview of different classes of phishing detection approaches, highlighting their strengths and weaknesses.

Table1
Comparison of phishing detection approaches

Approach	Strengths	Weaknesses	Typical Use Cases
Blacklist – based	Simple to implement, very fast, low overhead	Ineffective against new/short – lived domains, requires frequent updates	Web browsers, email filters
Heuristic rule – based	Can detect unknown threats, lightweight	Rules are static, easily bypassed by adaptive attackers	Intrusion detection systems (IDS
Classical ML models	Good accuracy, adaptable to new data	Limited interpretability, requires labeled datasets	Security monitoring, web gateways
Deep learning models	High detection rate, captures complex patterns	Black – box nature, high computational cost, risk of overfitting	Large – scale threat intelligence
Explainable AI (XAI)	Combines accuracy with interpretability	May be less accurate than deep models in some cases	Banking, e – government, enterprise SOC

3. Framework Design and Methodological Basis

The developed method implements a step-by-step processing of URL addresses, covering five key stages: validation and parsing; construction of a feature representation (portrait); classification; activation of a backup model in case of uncertainty; and formation of the final result with an explanation.

The initial stage encompasses a preliminary analysis module, or parser, whose primary function is to convert a URL from an arbitrary sequence of characters into a structured set of components. The input is segmented in accordance with the RFC 3986 standard into the following components: protocol, domain name, top-level domain (TLD), number of subdomains, path, query parameters, and risk markers. It is noteworthy that risk markers include special characters (@, %, =, –) or keywords (login, verify, token, update, email). During parsing, the syntax is also subject to validation to ensure its conformance to established standards. In the event that mandatory elements (e.g., top – level domain or protocol) are absent, the request is typically rejected.

In accordance with the study's methodology, a balanced sample of 1,000 addresses was compiled, with 500 allocated to each class (i.e., phishing/legitimacy). An analysis was then conducted to identify the most salient features. A significant finding of the study was that the majority of phishing addresses (83% of cases) utilized the HTTP protocol rather than the more secure HTTPS protocol. In contrast, only less than 7% of legitimate addresses employed the HTTPS protocol. Marker words such as "login," "verify," or "update" are present in more than 40% of

phishing records, but are almost never found in normal URLs. Furthermore, phishing domains have a propensity to utilize low-trust TLDs, such as .tk, .xyz, and .online, which are also considered in the generation of features.

The subsequent step involves the formation of a feature representation of the URL, which is interpreted in the work as an "address portrait." In this representation, all features are highlighted as numerical, logical, or a specific categorical parameter that reflects a specific characteristic of the URL. The method utilizes a total of 12 features, which are distributed across content categories. These categories include structural features, such as address length and the number of subdomains; technical features, such as the presence of HTTPS and TLD; semantic features, such as keywords; and morphological features, such as path length and query parameters. This selection of characteristic features enables the description of any address in vector format, irrespective of language, region, or platform.

The constructed portrait is then entered into a classification module, which determines the address's classification as either legitimate or phishing based on a two – level classification logic.

The initial level employs a logistic regression (LR) model, which estimates the probability of belonging to the phishing class by utilizing a function. This particular model was selected due to its explainability, which allows for the interpretation of results. Each feature is assigned a numerical weight, with the presence of a token increasing the probability of phishing by +0.17, the absence of HTTPS by +0.25, and so forth [13].

In instances where the outcome of the logistic regression is borderline (e.g., within the range of 0.45-0.55), the method initiates a secondary model known as GMDH (Group Method of Data Handling). This model employs a multilayer structure, incorporating internal feature selection and the formation of logical rules. In contrast to LR, which exhibits linear behavior, GMDH permits the existence of complex, nonlinear relationships between parameters. To illustrate, a potential rule might be formulated as follows: The following condition is indicative of a phishing attempt: "if tld = tk and has_https = 0 and contains_login = 1." Figure 2.1 provides a graphical representation of the described logic in the form of a UML diagram.

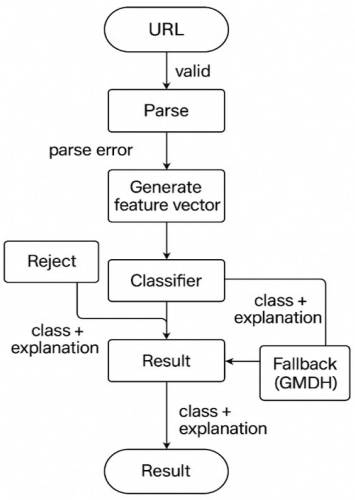

Figure 2: Information flow diagram of the URL classification method.

The resulting output is a JSON – type structure that contains not only the predicted class and confidence, but also an explanation based on weights or the GMDH tree. The output data can be stored in system logs with its integration, with further refinement of the method, allowing for decision auditing or error – based learning.

3.1. Structured Portrait of URLs

URL feature portraits are formed to unify input data and ensure its processing by classification algorithms. At this stage (Fig. 3), each address is converted into a feature vector that reflects its

structural and semantic characteristics, as well as additional attributes obtained from external sources. The present study employed a set of twelve key features: URL length (url_length), number of subdomains (num_subdomains), presence of HTTPS protocol (has_https), presence of keywords (contains_login, contains_token, contains_verify, contains_update, contains_email), path length (path_length), query length (query_length), number of "=" signs in the query (num_eq_signs), and top – level domain (tld_code). In order to enhance the informativeness of the models, the domain age feature was integrated, determined using WHOIS data (domain_age_days).

	url_length	num_subdomains	has_https	contains_login	contains_token	\
0	22	1	1	0	0	
1	23	1	1	0	0	
2	24	1	1	0	0	
3	21	1	1	0	0	
4	25	1	1	0	0	

	contains_verify	contains_update	contains_email	path_length	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	query_length	num_eq_signs	tld_code	domain_age_days
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Figure 3: Example of generated portraits based on the first 5 lines.

The procedure for constructing feature portraits is implemented using the Python programming language and the pandas, numpy, urllib.parse, and whois libraries. The algorithm is comprised of several processes. First, each Uniform Resource Locator (URL) is parsed. Then, the values of the corresponding features are calculated. Next, exceptional situations are handled (e.g., missing or incorrect WHOIS data) [10]. Finally, a single feature data frame is formed for further standardization and use in machine learning algorithms. This approach is designed to circumvent discrepancies in data structure and to ensure a uniform representation format for all examples.

3.2. Logistic Regression as a Transparent Classifier

In this study, we utilize logistic regression as the fundamental classification algorithm, operating on constructed feature portraits of URL addresses. In contrast to more intricate models, it facilitates not only the prediction of an object's class but also the elucidation of the rationale behind specific decisions, rendering this classification imperative in the context of enhancing the traceability of decisions. This is exemplified in scenarios such as the implementation of automatic request blocking or the generation of reports for security services.

The logistic regression model is based on the assumption that the class of the input address – phishing or not – can be predicted by analyzing the weighted sum of its features [13]. More recent explainable-AI studies further show that phishing detection can be enhanced by combining model transparency with LLM-powered interpretability [14], while transformer-based and LLM-based approaches confirm the importance of explainability in phishing-related detection tasks [15]. This sum is converted to a value between 0 and 1 using a sigmoid function, which is called logistic. If we denote the feature vector as $\vec{x}$ and the model weight vector as, then the model has the form represented by the following formula (1):

$$P(y=1|\vec{x})=\frac{1}{1+e^{(-(\vec{w}\,\vec{x}+b))}} \tag{1}$$

where $P(y=1|\vec{x})$ is the probability that the address is a phishing address;

$\vec{w}\,\vec{x}$ is defined as the scalar product of weights and features;

b is the bias.

During the training stage, these parameters are selected to minimize the loss function, which takes into account the classification error.

A distinguishing characteristic of this model in this context is its utilization of a vector comprising distinctly delineated features, including but not limited to URL length, the presence of "https," the existence of a token, the top – level domain (TLD), and the quantity of subdomains. The impact of these addresses on the result is determined by their degree of characteristic alignment with phishing addresses, exhibiting a positive or negative influence depending on this alignment. For instance, if the weight of the has_https feature is negative, its presence reduces the likelihood of phishing; if the weight of contains_login is positive, this feature increases the risk. Subsequent to the training of the logistic regression model on a balanced sample (Fig. 4), an analysis was conducted on the weight coefficients that determine the influence of each feature on the classification decision.

```
Result
[('has_https', -6.067393179729547),
 ('contains_login', 1.9767573447403564),
 ('tld_com', -1.3651550176874023),
 ('tld_br', 1.3375793694268692),
 ('tld_org', -1.2713711099047245),
 ('num_subdomains', -0.8867829451166926),
 ('tld_edu', -0.806977966536929),
 ('tld_ca', -0.6969267796250036),
 ('url_length', 0.6529362351238603),
 ('tld_nz', 0.5892287251233167),
 ('tld_cc', 0.48144825426441495),
 ('tld_co', 0.4776945114569775),
```

Figure 4: The logistic regression model has been trained on a balanced sample.

The construction of such a model necessitates that all feature vectors possess uniform length, a consistent scale of values, and be converted to a format that facilitates object comparison within the same space. The preceding sections thus implemented normalization, binarization, and categorical encoding to ensure the integrity of these conditions. The result is an input space where each point is a formalized representation of a real address, and logistic regression effectively "draws" a hyperplane that separates classes in this space.

3.3. GMDH for Nonlinear Relationship Discovery

The logistic regression implemented in the previous section has been demonstrated to achieve high levels of accuracy in the classification of URLs based on feature vectors. However, its linear nature imposes significant limitations, particularly in terms of its inability to adequately account for complex dependencies between features that only manifest themselves in their interaction. In scenarios where classes exhibit substantial overlap in the feature space or where features do not additively combine, there is a compelling need for a model that can independently establish an optimal structure, select relevant connections, and generate nonlinear dependencies.

In such conditions, it is advisable to employ a method that combines machine learning with internal structural evolution – the GMDH (Group Method of Data Handling) model [16], which belongs to the class of self-organizing systems. The method was proposed by academician Oleksiy Ivakhnenko in the 1960s and has since demonstrated its efficacy in scenarios where classical regression is too rudimentary and neural networks are excessively opaque. Consequently, this model is employed in the present project as the second tier of classification, that is, it is initiated when logistic regression yields an uncertain result or a low level of confidence (e.g., in the range [0.45;0.55]).

A distinctive attribute of GMDH is its capacity to autonomously formulate its structural configuration from multiple levels of nodes, with each node implementing a local model, most commonly in the form of a second – order polynomial. During the construction of the model, the method is employed to test various combinations of features, evaluate the quality of the model on a control sample, and gradually reject redundant or weakly informative connections. Consequently, GMDH facilitates an intelligent selection of features and structure without the need for external intervention.

3.4. Risk Landscape Analysis with SOM

In this study, a Self-Organizing Map (SOM) is employed to project high-dimensional vectors onto a two-dimensional space [17]. Addresses with similar characteristics are positioned in close proximity to one another, whereas those with differing characteristics occupy distinct regions of the map.

The SOM operates through competitive, unsupervised learning. Each URL is presented to the network and compared with the weight vectors of the neurons. Step by step, the Best Matching Unit (BMU) – the neuron whose weight vector most closely matches the input – is identified. The BMU and its neighboring neurons then adjust their weights toward the input vector, enabling the SOM to self-organize and produce a topologically ordered data representation.

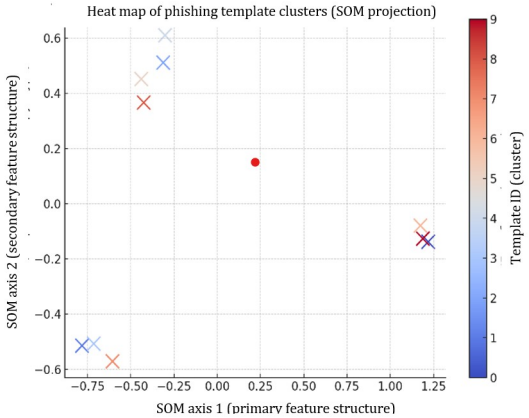

Figure 5: Heatmap of Phishing Pattern Clusters.

The map dimensions were determined experimentally by analyzing quantization and topographic error metrics. The most stable performance was observed with a 10 × 10 neuron configuration, which provided an effective balance between structural detail and generalization. Similar SOM-based and hierarchical clustering approaches have also demonstrated the usefulness of self-organizing representations for grouping complex objects according to latent feature similarities [18]. To reduce the risk of overfitting, the SOM was evaluated on a reference set of phishing URLs. This evaluation confirmed the map's capacity to capture risk structure even for addresses exhibiting novel or non-standard patterns [13].

As illustrated in Figure 5, the SOM space is expressed through the abstract coordinates "SOM Axis 1" and "SOM Axis 2," which are automatically generated during training to reflect object similarity. Each cross mark represents an SOM node that aggregates a group of URLs with comparable properties. Node color is determined by the cluster identifier assigned by the SOM based on feature-profile similarity. Warmer colors (e.g., red) indicate nodes where the SOM detected the most pronounced phishing patterns, demonstrating that even URLs with differing textual structures form stable clusters within the SOM space and reveal an underlying feature logic.

4. Experimental Results and Analysis

The comparative analysis presented in Table 2 of the results of the GMDH and logistic regression models allows for an evaluation of their effectiveness according to the main classification metrics. Additionally, it facilitates an exploration of the advantages and limitations of each model in the context of the task of detecting phishing URLs.

A thorough examination of the metrics reveals that both models exhibit high classification quality, with the GMDH model demonstrating a slight advantage in all indicators except precision, where there is a slight decrease (0.9965 vs. 0.9967). This outcome is consistent with the tendency of models to more effectively detect phishing attacks (recall), albeit with a slight rise in false positives. This finding indicates that the GMDH model exhibits a marginally superior capacity to detect all

instances of phishing (recall), a capability that is imperative for its practical implementation in the realm of cyber defense.

Table 2
Comprehensive comparison of key metrics

Metrics	Logistic Regression	GMDH (Polynomial)
Accuracy	0.9873	0.9880
Precision	0.9967	0.9965
Recall	0.9482	0.9517
F1 – score	0.9719	0.9736
ROC AUC Score	0.9885	0.9890

The elevated ROC AUC score (0.9890) of the GMDH model signifies its capacity to effectively differentiate URL categories across the full range of classification thresholds. This property is of particular relevance in scenarios where the cyber defense system is required to function in diverse sensitivity modes, such as "strict" or "moderate" modes, contingent on business requirements or traffic characteristics. Conversely, the ROC AUC score of the logistic regression model (0.9885) exhibited a negligible difference from the GMDH score and demonstrated slightly less flexibility in these scenarios.

With regard to the interpretability of decisions, logistic regression demonstrates a clear advantage due to the linear structure of the model, in which each feature is assigned a distinct weight coefficient, thereby facilitating the discernible impact of each feature on the risk of a phishing attack. This property is particularly useful for Explainable AI (XAI), where it is necessary to clearly justify each decision of the cyber defense system. For instance, the weight coefficient for the has_https feature can be directly interpreted as a risk factor or vice versa, depending on its sign.

The GMDH model with second – order polynomial nodes offers enhanced interpretability due to its capacity to accommodate interactions among features. This is particularly salient in scenarios where phishing attack patterns do not manifest as a consequence of a singular feature, but rather, they are constituted by the amalgamation of multiple factors (e.g., has_https × num_subdomains). These interactions are frequently exploited by cybercriminals to circumvent filtering mechanisms. The GMDH weight coefficient graphs demonstrate the impact of individual features on the model, with these features being evaluated in pairs.

The authors confirm that no generative artificial intelligence tools were used in the conception, drafting, or editing of this research manuscript. All methodological design, experimental results, analysis, and textual content were produced exclusively by the human authors. Any use of software pertains solely to standard academic tools for data processing (e.g., Python, pandas, scikit-learn) and document preparation, with no involvement of AI-based text generation or content synthesis systems.

5. Conclusions and Future Perspectives

An analysis of the matrix presented in Figure 4 reveals that the value 9.997e+01 (nearly 100%) in multiple nodes signifies the presence of nodes where phishing URLs predominate. This approach enables the identification of the so – called SOM "hot spots," defined as nodes where the accumulation of risk is maximized. These nodes necessitate particular consideration during the monitoring of the cyber defense system, as they may indicate a significant degree of threat.

Conversely, nodes with a value of zero or near zero indicate an absence or minimal level of phishing activity. These clusters can be regarded as "safe" due to the predominance of legitimate addresses and data that do not inherently pose significant risks. Consequently, visualizing this matrix facilitates not only the identification of risky nodes but also the establishment of the

foundation for automatically constructing security policies, such as the implementation of a blocking mechanism for addresses in nodes exhibiting a risk level exceeding 90%.

```
[[9.971e+01 9.998e+01 0.000e+00 1.000e+02 9.995e+01 3.300e-01 0.000e+00
  9.997e+01 6.800e+00 8.240e+00]
 [0.000e+00 0.000e+00 1.380e+00 0.000e+00 2.948e+01 0.000e+00 0.000e+00
  0.000e+00 0.000e+00 8.980e+00]
 [0.000e+00 7.060e+00 0.000e+00 8.000e-02 0.000e+00 2.600e-01 0.000e+00
  9.915e+01 0.000e+00 0.000e+00]
 [0.000e+00 5.180e+00 0.000e+00 1.000e+02 2.500e-01 5.900e-01 0.000e+00
  0.000e+00 0.000e+00 0.000e+00]
 [0.000e+00 0.000e+00 1.320e+00 0.000e+00 1.000e+02 0.000e+00 0.000e+00
  7.600e-01 1.000e+02 3.700e-01]
 [3.240e+00 3.060e+00 1.250e+00 1.440e+00 1.000e+02 9.996e+01 1.000e+02
  9.600e-01 0.000e+00 2.300e-01]
 [1.330e+00 2.830e+00 0.000e+00 0.000e+00 9.998e+01 0.000e+00 9.706e+01
  0.000e+00 2.800e-01 1.000e+02]
 [2.170e+00 5.417e+01 4.100e-01 1.400e-01 0.000e+00 9.878e+01 0.000e+00
  2.800e-01 8.771e+01 1.370e+00]
 [0.000e+00 1.461e+01 0.000e+00 0.000e+00 8.667e+01 0.000e+00 1.000e+02
  7.200e+00 6.400e-01 9.997e+01]
 [9.070e+00 7.400e-01 0.000e+00 9.526e+01 8.270e+01 0.000e+00 1.210e+00
  3.800e-01 0.000e+00 9.997e+01]]
```

Figure 4: The following figure illustrates the distribution of the percentage of phishing attack risk in each SOM node, as determined by the results of the LR and GMDH models.

Consequently, the study enabled SOM to identify nodes with a high concentration of phishing patterns, thereby facilitating the generation of heat maps that illustrate risk and highlight potential areas of vulnerability for users. The employment of logistic regression and GMDH models has been demonstrated to yield high classification accuracy, with an F1 – score exceeding 97%, thereby substantiating their efficacy in detecting phishing attacks. A statistical analysis of features was conducted, enabling the identification of the key characteristics of URLs that most influence classification results. This analysis also demonstrated the importance of a multifactorial approach to building a detection system.

The findings of the study substantiated that the incorporation of SOM with the mechanism for calculating the average risk in nodes facilitates not only the grouping of URLs by features, but also the detection of latent high – risk patterns, thereby considerably enhancing the accuracy of phishing attack detection. The proposed mechanisms for local classification refinement and support for autonomous analysis via CLI demonstrate the flexibility of the method and its ability to adapt to real – world usage scenarios.

The developed architecture of the method has proven its competitiveness due to its ability to effectively analyze both static data (offline) and current traffic (online), as well as the ability to dynamically update knowledge in integrated systems based on user feedback.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2023. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[2] CERT – UA Reports: 11 Ukrainian Telecom Providers Hit by Phishing Attacks. https://thehackernews.com/2023/10/cert-ua-reports-11-ukrainian-telecom.html

[3] M. Bitaab, H. Cho, A. Oest, et al. Scam Pandemic: How Attackers Exploit Public Fear through Phishing, 2021. https://arxiv.org/abs/2103.12843.

[4] S. Toliupa, S. Buchyk, A. Shabanova, O. Buchyk (2023) The Method for Determining the Degree of Suspiciousness of a Phishing URL. In 10th International Scientific Conference "Information Technology and Implementation" (IT&I–2023), Kyiv, Ukraine, November 20-21, 2023, pp. 239-247.

[5] B. Gupta, N. Arachchilage & K. Psannis, Defending against Phishing Attacks: Taxonomy of Methods, Current Issues and Future Directions, 2017. https://arxiv.org/abs/1705.09819.
[6] D. Zhuravchak, M. Opanovych, A. Tolkachova, V. Dudykevych, A. Piskozub (2024) Design of an Integrated Defense-in-Depth System with an Artificial Intelligence Assistant to Counter Malware. *EasternEuropean J. Enterp. Technol.*, 6(2(132)), pp. 64-73. doi:10.15587/1729-4061.2024.318336.
[7] O. Partyka, A. Partyka, A. Imoize (2026) AI for Web Application Security. *Handb. Cybersecur. Challenges Solutions Emerg. Technol.* doi:10.1201/9781003640790-14.
[8] D. Lain, K. Kostiainen & S. Capkun, Phishing in Organizations: Findings from a Large – Scale and Long – Term Study. 2021. https://arxiv.org/abs/2112.07498.
[9] I. Opirskyy, O. Harasymchuk, O. Mykhaylova, Y. Sovyn, I. Tyshyk (2025) Detection of Network Attacks Based on the RedHunt Virtual Machine. *Adv. Cybersecurity: NextGeneration Syst. Appl.* Boca Raton: CRC Press, 2025, pp. 310-330. doi:10.1201/9781003546153-14.
[10] H. Hulak, L. Kriuchkova, P. Skladannyi, I. Opirskyy (2021) Formation of Requirements for the Electronic Record-Book in Guaranteed Information Systems of Distance Learning. *CEUR Workshop Proc.*, Vol. 2923, pp. 137-142.
[11] A. Ghazi – Tehrani, & H. Pontell, Phishing Evolves: Analyzing the Enduring Cybercrime. Vict. & Offender, 16(3), 316-342, 2021. https://www.utica.edu/academic/institutes/cimip/Phishing_Evolves_Analyzing-the-Enduring-Cybercrime.pdf
[12] V. Khoma, D. Sabodashko, V. Kolchenko, P. Perepelytsia, M. Baranowski (2024) Investigation of Vulnerabilities in Large Language Models Using an Automated Testing System. *CEUR Workshop Proc.*, Vol. 3826, pp. 220-228.
[13] V. Vajratiya, B. Gupta, A. Gaurav Mutual information based logistic regression for phishing URL detection. Cyber Secur. Appl., 2, 100044. doi:10.1016/j.csa.2024.100044.
[14] B. Lim, R. Huerta, A. Sotelo, A. Quintela, P. Kumar EXPLICATE: Enhancing phishing detection through explainable AI and LLM – powered interpretability. https://arxiv.org/abs/2503.20796.
[15] M. Uddin, I. Sarker I. An explainable transformer – based model for phishing email detection: A Large Language Model approach. arXiv preprint, 21.02.2024. https://arxiv.org/abs/2402.13871.
[16] O. Dag, Yozgatligil C. (2016) Architecture of the GMDH algorithm: Principles and layer – wise design. R J. *(2016) 8:1, pages 379-386.* doi:10.32614/RJ-2016-028.
[17] A. Guérin, P. Chauvet, F. Saubion. A survey on recent advances in Self – Organizing Maps. *arXiv*, 2024. https://arxiv.org/abs/2501.08416.
[18] Y. Hawas, M. Al – Shaher, A novel deep SOM and hierarchical clustering for content recommendation. *BIO Web of Conferences, 97*, 00088. doi:10.1051/bioconf/20249700088.