TSARKA SCIENCE

Volume 1 (2026), Issue 2
pp. 86–97

eISSN: 3134-6057
pISSN: 3134-6049

DOI: 10.66571/tsarka-3134-6057-09

ANALYSIS AND IMPROVEMENT OF MACHINE LEARNING METHODS FOR DETECTING MALICIOUS LINKS ON INSTAGRAM

A. Ibraikhan[1*], Zh. Tashenova[1]

[1]L.N. Gumilyov Eurasian National University, Astana, Kazakhstan
*Corresponding author: ibraikhann@gmail.com.

Abstract

The distribution of malicious URLs on social network websites, especially Instagram, is a serious cybersecurity risk to end-users. Although machine learning methods have enhanced the threat detection process, existing literature is often afflicted with a critical methodological issue, which is the lack of domain isolation when evaluating the model. This lack of isolation will result in data leakage and artificially inflated performance metrics that cannot be generalized to zero-day attacks. In this paper, the gap is filled by providing a powerful malicious link detection framework. In order to achieve complete objectivity, we also apply strict domain-split cross-validation strategy (Group Shuffle Split), which is very effective in removing data leakage vulnerabilities. In addition, the paper compares two developed, independent architectures, one, a character-based 1D Convolutional Neural Network (CNN) used to perform automated pattern recognition, and another, a Hybrid Ensemble system. The proposed ensemble is able to combine deep semantic embeddings of Large Language Models (DistilBERT) with classical gradient-boosting algorithms with the help of a Soft Voting mechanism, reinforced by a lexical and structural feature engineering strategy. This approach greatly decreases the false-positive rates, as well as prevents the trivial memorization of host identifiers. Lastly, to illustrate a realistic application, the theoretical frameworks are converted into a working functional prototype, including a REST API and a client-side browser extension, which is dedicated to the proactive, real-time protection of social media users.

Keywords: *Malicious URLs, Cybersecurity, Convolutional Neural Network (CNN), Hybrid Ensemble, DistilBERT, Group Shuffle Split, Data Leakage, Instagram.*

1. Introduction

Social media has already found its niche in the contemporary society, and it drastically changes the fundamental paradigm of ordinary interaction and business operations. However, this increased rate of digitalization is accompanied by severe neg-

Volume 1 (2026), Issue 2
pp. 86–97

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

ative outcomes, including the widespread availability and proliferation of cybercrime [1]. The use of malicious hyperlinks is one of the major vectors of attack in this cyberspace. Malicious actors actively use URLs as landing pages to conduct phishing campaigns and spread malware into the systems of users quietly, endangering the personal data and financial resources of millions of users [2,3].

Conventional cybersecurity systems are now experiencing a serious efficacy crisis, with any defense strategy that is based entirely on fixed blacklists no longer being sufficient [4]. In order to avoid such filters, cybercriminals constantly develop their methods and use URL-shortening services and the mass production of new domain names (DGA) to hide the real addresses of destinations. Traditional filtering systems do not have the ability to evolve with this very dynamic threat environment, and therefore, as a rule, fail to identify zero-day attacks [5].

In order to overcome these weaknesses, machine-learning (ML) solutions have remained a rapidly adopted concept in academia. Models that jointly examine the lexical structure of URLs and host level characteristics show much higher results than traditional heuristic methods [6,7]. The critical analysis of the literature available however shows a conceptual gap in methodology that occurs recurrently. One of the most prominent issues is the insufficient attention to model generalization: most studies use naive cross-validation schemes (e.g. random splits) that do not take into consideration domain-based grouping, which inevitably results in data leakage during the training phase. As a result, these models generate artificially exaggerated measures of accuracy in situations of controlled conditions but do not scale and project to real-world operational cases. Moreover, the traditional ML algorithms can only be useful in extracting simple heuristic features, but they cannot model deep sequential dependencies within the character structures of a URL, a task that modern deep neural network architectures are more adept [8,9].

The given paper suggests a strict rationalization of the security-system development process, which is concerned with objective assessment and architectural strength. The problem of data leakage is thoroughly overcome by applying GroupK-Fold cross-validation scheme that introduces strict domain segregation. The problem of imbalance and bias in the distribution of classes is solved by incorporating authoritative URLs of the whitelists that are highly reputable. Methodologically, the research constructs and tests two independent ML paradigms, which are, first, a special one-dimensional Convolutional Neural Network (1D CNN) that learns to extract latent structural patterns of character-level sequences automatically, and second, a hybrid ensemble system that trains the deep semantic processing capabilities of Large Language Models (DistilBERT) and combines them via soft voting.

The key contributions of this paper are:

- Eradication of Data Leakage: The fundamental requirement of applying a GroupK-

TSARKA SCIENCE
Volume 1 (2026), Issue 2
pp. 86–97
eISSN: 3134-6057
pISSN: 3134-6049

Fold strategy to disentangle domain names in cross-validation has been experimentally established, which has been capable of preventing overfitting and delivering a fair evaluation of the resilience of the model to zero-day attacks.

- Architecture of Advanced Machine Learning Systems: The paper presents and compares two very successful architectures, one a character-level CNN to identify sequence patterns, and the other a robust hybrid architecture, which uses semantic token embedding and standard algorithms.
- Distribution Bias Mitigation: A strategic inclusion of authoritative URLs of high-reputation whitelists was effectively used to anchor the priors of the model, and this contributed greatly to robustness and greatly reduced the false-positive rate.
- Live Implementation and Deployment: The theoretical and mathematical findings of the study have been converted into an operational proof-of-concept, containing a successful REST API back-end and a client-side browser extension that are meant to deliver proactive and real-time protection to end users.

2. Material and research methods.

This research gives more emphasis on methodological transparency especially on technical implementation. The pipeline of analytical tasks, which includes the processing of raw data to the final assessment, has a number of individual steps. The overall approach is a comparative one and classical machine learning models are used to consider the proposed neural network and hybrid ensemble systems as well as compare their performance to that of these models.

2.1. Data Preparation and Preprocessing

The Malicious URLs experimental framework is based on the open-source Malicious URLs dataset [10]. There are hundreds of thousands of various threat samples such as phishing, malware, and defacement URLs as well as legitimate links in this corpus. The raw string data however needs a lot of preprocessing before it can be successfully used by the machine learning algorithms. The data curation and cleaning process was further broken down into three major steps:

First, an external whitelist of high-reputation domains, the Majestic Million [11] was added to reduce the false-positive rate, which is a very important requirement in contemporary detection systems. Most importantly, the dataset was deliberately left in its inherently imbalanced form (the legitimate URLs making up about 67.4% of the corpus) in order to mirror the actual distributions of the internet traffic. Instead of using destructive undersampling methods, which cast out useful data and distort the existing probabilities, the inherent imbalance was also handled algorithmically by adjusting the weights of the classes in the training stage.

Second, there was a large bias in the raw data, with some individual hosts being

Volume 1 (2026), Issue 2
pp. 86–97

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

overrepresented with thousands of associated links. This skew may make a model memorize certain domains instead of acquiring the structural patterns of malicious URLs. In order to overcome this risk, a hard limit was put on the number of URLs per host, by restricting the dataset to a maximum of 50 URLs per host.

Third, since the classical machine learning models are incapable of accepting raw text as input, the URLs were converted into numerical feature vectors [12]. In order to avoid the bias in the datasets and spurious correlations a Component-aware Feature Extraction strategy was adopted. Rather than trying to isolate primitive characteristics on the whole URL string, 18 manual heuristic characteristics were rigidly separated into two vectors, domain-level characteristics (e.g., domain length, the presence of an IP address, number of subdomains) and path-level characteristics (e.g., path depth, the presence of suspicious file extensions, or base64 encoding). This component isolation is to make sure that the algorithms can assess host authority and path semantics in isolation of each other.

2.2. Baseline Approach: Classical Machine Learning Models

This study needed a good comparative baseline that was established. Gradient-boosting models and decision tree ensembles were selected as their performance has been extensively proven effective when working with structured tabular data [6,13]. Namely, LightGBM, XGBoost, and Random Forest models were utilized. These algorithms are described as having high computational efficiency and having high abilities to process complex numerical feature vectors.

The preprocessed manual heuristic features were used to initially train them and their performance in prediction was the bar of quality that the later neural network architectures were required to exceed (Fig. 1). Moreover, instead of acting merely as a reference point for comparison, the mathematical machinery of all these classical algorithms was later unified to become the backbone of the eventual hybrid classification ensemble.

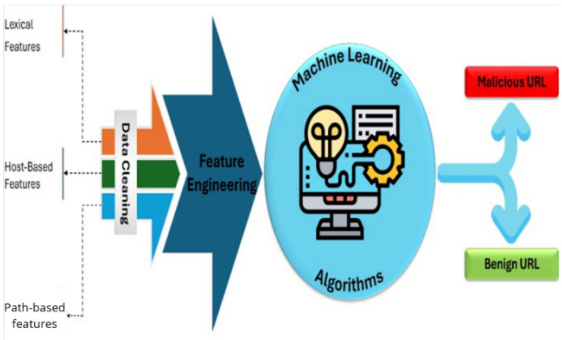


Figure 1. The conceptual pipeline of feature extraction and classification by classical machine learning algorithms

TSARKA SCIENCE
Volume 1 (2026), Issue 2
pp. 86–97
eISSN: 3134-6057
pISSN: 3134-6049

2.3. A Better Strategy: Neural Network and Ensemble Techniques

Identification of the latent patterns within the URL address structure involves the application of advanced computational paradigms that are impossible to attain using the conventional algorithms. Although classical methods of machine learning may be very effective in situations where the data is organized and numerical in nature, they are highly insufficient when it comes to eliciting profound semantics of unstructured textual data. The scientific contribution of this study is the qualitative enhancement of the classification scheme through the use of two independent advanced models: a character-level Convolutional Neural Network (1D CNN) and a hybrid ensemble, which combines Large Language Models (DistilBERT) and gradient boosting algorithms using soft voting.

2.3.1. Convolutional Neural Network (CNN) Architecture

To compare the performance of feature extraction at the character level, the 1D CNN architecture is used as a baseline model. As opposed to traditional parallel structures where a web address is perceived as a combination of autonomous heuristic properties, the CNN architecture treats a web address as an array of unsubstitutable characters. Such a choice of methodology allows the model to reflect local dependencies and structural context of the sequence [9] (Fig. 2). The present-day software implementation of this baseline model was conducted with the PyTorch framework in order to achieve the high computational performance in the analysis of string data. The resulting structure is a deep topology, in which the successive layers are particular to some feature-extraction processes:

1. Embedding Layer: The mapping of each of the individual hyperlink characters into a high-density vector of fixed dimension is performed in the first phase. This kind of transformation also offers the network the capability to construct a mathematical space that shows the semantic proximity of dissimilar characters and their combinations [14].
2. Convolution Layers (Conv1D): This is then successively followed by parallel convolutional layers of different sizes. The layers are used as character n-gram detectors when working on URL analysis [8]. They scan the text sequence in parallel and, consequently, they discover bits of a string which can be described as a pointer of phishing and malware (login, secure, or php).
3. Global Pooling Layer (GlobalMaxPooling1D): The dimensionality of the data grows exponentially with the process of multi-channel convolution operations and so a global max-pooling layer is employed. It also eliminates the irrelevant noise and only the highest saliency (maximum) feature activations are retained and projected into a small, unit-dimensional fixed-size vector.
4. Fully Connected Layers (Dense Layers): The last phase in the architecture is made up of fully connected dense layers the purpose of which is to do the ulti-

Volume 1 (2026), Issue 2
pp. 86–97

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

mate aggregation of the extracted patterns. This is further enhanced by the introduction of the Rectified Linear Unit (ReLU) activation function which gives the calculating process the non- linearity required and hence enabling the model to handle difficult class-separation tasks. The final layer (that employs the Softmax activation function) is the last one that provides the end probability distribution that provides the information on whether a URL belongs to the respective threat categories.

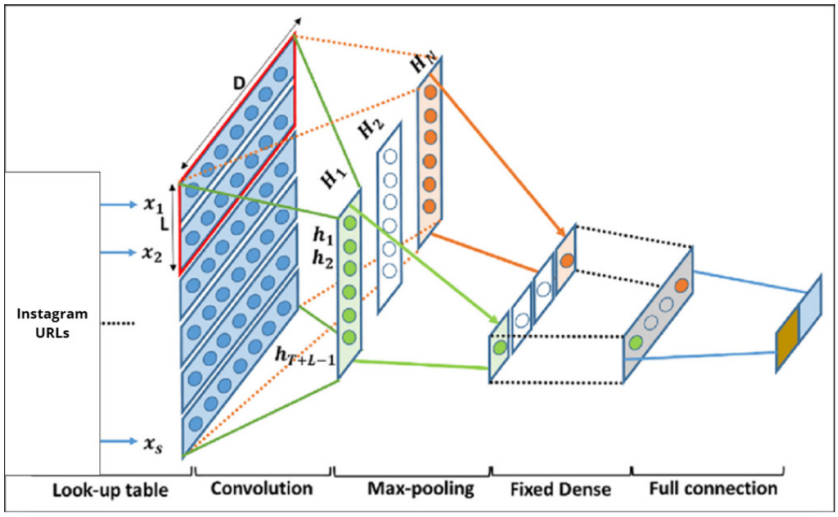


Figure 2. CNN on text classification [15]

2.3.2. Soft Voting Hybrid Ensemble Architecture

The combination of heterogeneous predictive algorithms enables a major improvement in the general predictive capabilities of a system by offsetting the individual shortcomings of separate models. In order to put this idea into practice, instead of depending on a more conventional multi-level stacking method, the present study will employ a Hybrid Ensemble framework which is founded on a Soft Voting mechanism. The methodology is consistent with the recent developments in the efficient ensemble techniques of cybersecurity [16]. The architecture proposed works on the following built-in components:

- Semantic Feature Extraction (DistilBERT): Operating parallel to the character-level CNN, this approach utilizes a pre-trained Large Language Model (DistilBERT). The model takes the URL strings and processes them into deep semantic embeddings (dense vectors), which capture the intricate contextual and lexical relations in the web address.
- Base Predictive Models: The obtained semantic embeddings are combined with the standalone manual heuristic features. This feature combination is then fed

TSARKA SCIENCE
Volume 1 (2026), Issue 2
pp. 86–97
eISSN: 3134-6057
pISSN: 3134-6049

into a diverse collection of powerful decision tree and gradient boosting algorithms namely LightGBM, XGBoost and Random Forest. All of these models are independent of each other and each is analysing the rich feature space to produce probability estimates of the corresponding threat classes.

- Aggregation through Soft Voting: The terminal prediction is performed using a Soft Voting Classifier, rather than using a secondary meta-classifier (e.g. logistic regression). This mechanism is an algorithmic computation of the average of the forecasted probabilities of the individual base models. The URL is then mapped to the threat category that has the highest mathematically weighted probability to produce a very accurate and computationally efficient final decision.

2.4. Process Strategy of Validation and Evaluation

Another phase of the research that is of utmost significance is the use of a rigorous validation measure to ensure a realistic and objective evaluation of the models' generalization. The traditional random sampling, which is used in the analysis of web addresses, inevitably results in gross methodological errors and data leakage; therefore in this study strict group cross-validation (Group Shuffle Split) strategy is used.

The tight data clustering was done according to the root domain so that all the URLs generated by the same host appear only in either the training set or the test set (zero overlapping domains). By doing this, such a methodology entirely eliminates the threat of data leakage and makes the algorithm unable to trivially memorize domain names, compelling the system to discover the underlying structural and semantic patterns of risks.

The models were quantitatively evaluated using standard classification metrics: Accuracy, Precision, Recall, Macro F1, and Weighted F1-scores. In the analysis, special emphasis was placed on the balance between precision and recall (Precision-Recall Trade-off). False positives, blocking out legitimate, valid material rather than a real cyber threat is too expensive and results in a very poor user experience in the context of end-user cyber threat detection. Thus, the architectural design and class weight balancing specifically prioritize minimizing and balancing of the class weights.

3. The results and discussion

This section describes the empirical findings, and evaluates the proposed system for detecting malicious URLs. The implementation of the experiments was done in Python, using Apple Metal Performance Shaders (MPS). Conventional models were trained using Scikit-learn, LightGBM, and XGBoost and the 1D CNN using PyTorch.

3.1 Feature Importance and Data Leakage

Exploratory research using a baseline LightGBM model showed that network location and regional URL hashes were the most important features for the model (Fig. 3). But on further inspection, it was found that the features result in high data leakage.

Volume 1 (2026), Issue 2
pp. 86–97

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

Rather, the model was simply memorising host ID labels from the training data. As such, features based on hashing were removed in the final design. Only safe lexical and structural characteristics (e.g., URL length, special characters, IP addresses in URL) were retained to enhance the model's zero-day threat protection.

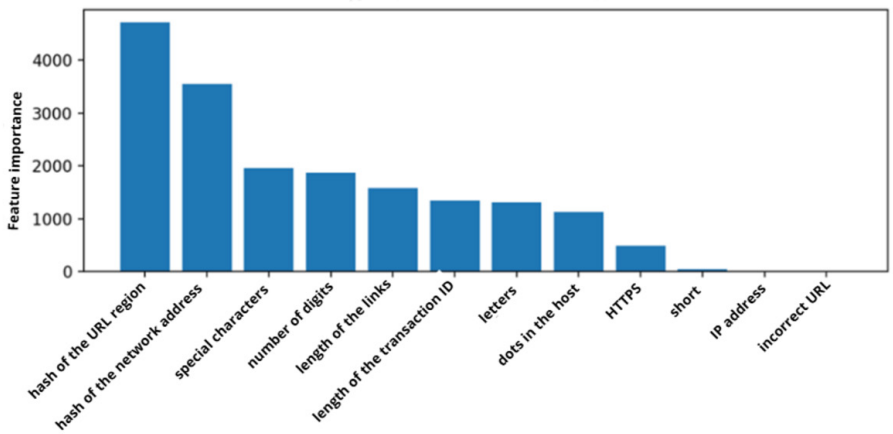


Figure 3. LightGBM: illustrating the data leakage vulnerability caused by hash-based features

3.2 Model Training Dynamics

The baseline gradient-boosting models (Fig. 4) revealed a steep decline in validation loss in the early stages of training, where it plateaued around 100 trees. However, the loss then began to rise - a sign of overfitting. This insight provided the basis for model generalization, by incorporating an Early Stopping technique into the final model pipeline.

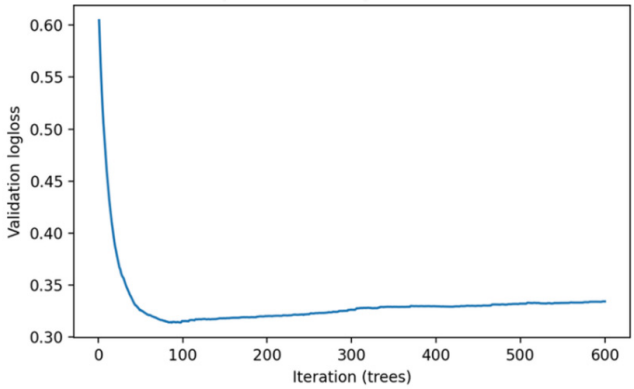
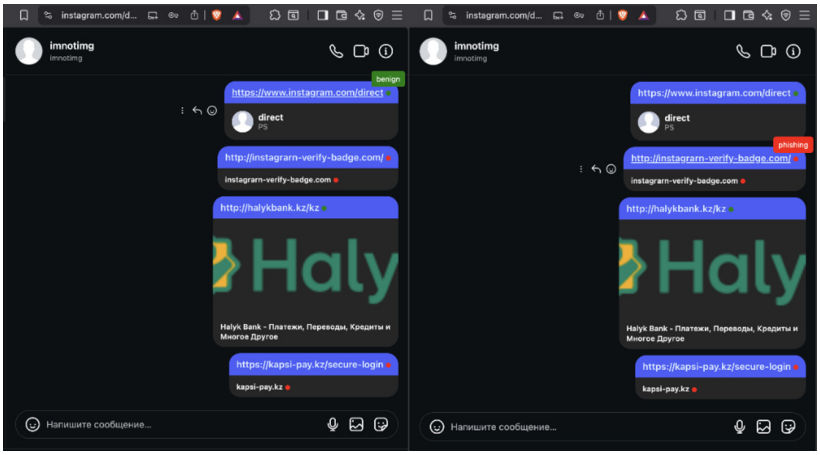


Figure 4. LightGBM learning curve

TSARKA SCIENCE
Volume 1 (2026), Issue 2
pp. 86–97
eISSN: 3134-6057
pISSN: 3134-6049

3.3 Proof-of-Concept Deployment

To prove the concept, the developed models were deployed via a browser extension. Figure 5 demonstrates the tool's proactive protection by successfully rating Instagram links and visually alerting the user to the security level of the link (green for benign, red for phishing) on the social network.



(a) (b)

Figure 5. Real-time threat detection: (a) Detection of a benign URL with a green indicator. (b) Detection of a phishing URL with a red indicator

3.4 Quantitative Performance Evaluation

The proposed models were compared with existing approaches using Accuracy, Macro F1 and Weighted F1 (Tab. 1).

Table 1. Comparative performance metrics and inference times

Model	Accuracy	Weighted F1	Macro F1	Inference Time
1	2	3	4	5
Random Forest (Baseline)	0.796	0.809	0.726	~5 ms
XGBoost	0.823	0.831	0.777	~5 ms
LightGBM	0.841	0.847	0.795	~5 ms
Character-level 1D CNN	0.856	0.861	0.829	~1 ms
Hybrid Ensemble (Soft Voting)	0.856	0.857	0.803	~150 ms

Volume 1 (2026), Issue 2
pp. 86–97

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

The best accuracy obtained with the baselines was 0.841. The character level 1D CNN and the proposed Hybrid Ensemble show much better results with an accuracy of 0.856.

Although the CNN was very efficient and had a marginally better Macro F1 score, a class-wise analysis of the results confirms the strategic advantage of the Hybrid Ensemble. Leveraging the semantic knowledge learned by DistilBERT, the hybrid approach led to a substantial improvement in the recall of the benign class to 0.9035. This confirms our hypothesis: the hybrid system significantly decreases the false alarm rate on complex benign URLs making it a promising solution for social media.

3.5 Discussion of Findings

- Feature vulnerabilities: As shown, the use of hash-based features leads to data leakage. Models should use Component-aware Feature Extraction and Early Stopping to prevent trivial learning and overfitting.
- Deep Learning Weaknesses: The isolated 1D CNN is a powerful tool to work with the raw character sequences but may mistake valid, complex URLs as malicious. The addition of DistilBERT to the Hybrid Ensemble overcomes this limitation.
- Strict Validation: Fair results can only be obtained by preventing data leakage during evaluation. This research uses the Group Shuffle Split, which guarantees domain separation and thus a true reflection of the system's capability to identify zero-day attacks.

4. Conclusion

In this paper, we described a systematic approach to the creation of an effective structure that can detect malicious URLs. One of the main goals of this study was to address the most important methodological shortcomings of similar research, as the validity of the reported performance metrics is often compromised by them. A domain-split cross-validation strategy (Group Shuffle Split) was followed strictly to ensure complete objectivity and the simulation of zero-day threat conditions. This rigorous methodology practically eliminated the vulnerabilities of data leakage, which are rampant and cannot be controlled when using the conventional random splitting tools.

The study used two advanced and independent architectures to carry out a thorough analysis of hyperlink data, namely a character-level Convolutional Neural Network (1D CNN) and a Hybrid Ensemble system. The suggested ensemble was able to integrate the intensive semantic processing of Large Language Models (DistilBERT) with conventional gradient-boosting and decision tree models through a Soft Voting system. Moreover, the adoption of a Component-aware Feature Extraction strategy also ensured that the models considered the host authority and path semantics as separate variables without spurious correlations.

TSARKA SCIENCE

Volume 1 (2026), Issue 2
pp. 86–97

eISSN: 3134-6057
pISSN: 3134-6049

The practical application of this system is grounded in the empirical findings. The theoretical and mathematical results of this work have already been converted into a proof-of-concept implementation, including a REST API back-end and a client-side browser extension to provide real-time detection of threats on social media. The next stage in research is scaling this deployment, performing a thorough error analysis to reduce false-positive rates on complicated URLs even more, and extending the compatibility of this system to the defense of end-users of more modern social media ecosystems.

References

1. Mulahuwaish, A.; Qolomany, B.; Gyorick, K.; Abdo, J.B.; Aledhari, M.; Qadir, J.; Carley, K.; Al-Fuqaha, A. A survey of social cybersecurity: Techniques for attack detection, evaluations, challenges, and future prospects. Computers in Human Behavior Reports 2025, 18, 100668. https://doi.org/10.1016/j.chbr.2025.100668.

2. Naz, A.; Sarwar, M.; Kaleem, M.; Mushtaq, M.A.; Rashid, S. A comprehensive survey on social engineering-based attacks on social networks. International Journal of ADVANCED AND APPLIED SCIENCES 2024, 11, 139–154. https://doi.org/10.21833/ijaas.2024.04.016.

3. Almohaimeed, M.; Albalwy, F.; Algulaiti, L.; Althubyani, H. Phishing URL detection using Deep Learning: A resilient approach to mitigating emerging cybersecurity threats. Ingénierie des systèmes d information 2025, 15, 1219–1227. https://doi.org/10.18280/isi.300510.

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

5. Tian, Y.; Yu, Y.; Sun, J.; Wang, Y. From past to present: A survey of malicious URL detection techniques, datasets and code repositories. Computer Science Review 2025, 58, 100810. https://doi.org/10.1016/j.cosrev.2025.100810.

6. Abad, S.; Gholamy, H.; Aslani, M. Classification of malicious URLs using machine learning. Sensors 2023, 23, 7760. https://doi.org/10.3390/s23187760.

7. Coste, C.I. Malicious Web Links Detection - A comparative analysis of machine learning algorithms. Studia Universitatis Babes,-Bolyai Informatica 2023, 68, 21–36. https://doi.org/10.24193/subbi.2023.1.02.

8. Le, H.; Pham, Q.; Sahoo, D.; Hoi, S.C.H. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. arXiv (Cornell University) 2018. https://doi.org/10.48550/arxiv.1802.03162.

9. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for

TSARKA SCIENCE

eISSN: 3134-6057
pISSN: 3134-6049

text classification. arXiv (Cornell University) 2015. https://doi.org/10.48550/arxiv.1509.01626.

10. Malicious URLs dataset, 2021.

11. The Majestic Million, 2026.

12. Aljabri, M.; Alhaidari, F.; Mohammad, R.M.A.; Mirza, S.; Alhamed, D.H.; Altamimi, H.S.; Chrouf, S.M.B. An assessment of Lexical, Network, and Content-Based features for detecting malicious URLs using machine learning and deep learning models. Computational Intelligence and Neuroscience 2022, 2022, 1–14. https://doi.org/10.1155/2022/3241216.

13. Maftoun, M.; Shadkam, N.; Komamardakhi, S.S.S.; Mansor, Z.; Joloudari, J.H. Malicious URL Detection using optimized Hist Gradient Boosting Classifier based on grid search method, 2024.

14. Saxe, J.; Berlin, K. eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys. arXiv (Cornell University) 2017. https://doi.org/10.48550/arxiv.1702.08568.

15. Nguyen, V. Q., Anh, T. N., & Yang, H. (2019). Real-time event detection using recurrent neural network in social sensors. International Journal of Distributed Sensor Networks, 15(6), 155014771985649. https://doi.org/10.1177/1550147719856492

16. Omolara, A.E.; Alawida, M. DaE2: Unmasking malicious URLs by leveraging diverse and efficient ensemble machine learning for online security. Computers Security 2024, 148, 104170. https://doi.org/10.1016/j.cose.2024.104170.

Information about authors

Ibraikhan Alinur Ruslanuly – Master's degree, Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

e-mail: ibraikhann@gmail.com

ORCID: https://orcid.org/0009-0003-0659-3898

Zhuldyz Tashenova Musagulovna – PhD, Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

e-mail: Zhuldyz_tm@mail.ru

ORCID: https://orcid.org/0000-0003-3051-1605