Hybrid Model for Detecting Fraudulent Domain Names*

Serhii Buchyk[1,*,†], Anastasiia Shabanova[1,†], Oleksandr Buchyk[1,†], Viktoriia Shmatko[1,†] and Maksym Kotov[1,†]

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

Abstract

The relevance of the study is related to the risks arising from the growing number of botnets that use domain generation algorithms (DGA) to avoid detection. The use of DGA makes it difficult to identify malicious servers, creating significant challenges for cyber defense. Traditional machine learning methods require manually created domain features that become ineffective when attack patterns change. This paper proposes a hybrid machine learning model for DGA domain detection that combines high adaptability and accuracy.

Keywords

botnet, domain generation algorithm, DGA domains, cybersecurity, machine learning, hybrid model

1. Analyzing threats and defense methods

In the second half of 2024, Alibaba's network became the leader in terms of the number of active C&C botnets, with 172 such servers, while Tencent ranked second with 85 active botnets, as shown statistically in Fig. 1.

Despite the reduction of botnet activity on 12 out of 20 networks, this is not enough to ensure cybersecurity, as even one unprotected C&C server can become a springboard for large-scale attacks. Networks located in China continue to be the main hubs of botnets: more than 60% of all active C&Cs operate there. The emergence of new entrants in the ranking is particularly critical, including cloudinnovation.org (#3), changway.hk (#10), and ctgserver.com (#11).

The insufficient response of hosting operators to reports of abuse indicates the low effectiveness of existing mechanisms to combat botnets. Neglecting such problems not only damages the reputation of these companies but also allows attackers to continue their activities without significant obstacles. Even noticeable reductions, such as minus 77% for hetzner.com or −13% for ucloud.cn, are only a partial solution, as they do not cover the full picture of cyber threats [1].

Cobalt Strike continues to hold the lead among malware associated with botnet command and control (C&C) servers. Its popularity is attributed to its versatility, ease of use, and ability to mimic the actions of legitimate users on the network, making it much more difficult to detect. A 12% increase in the second half of 2024 underlines the steady demand for this tool among cybercriminals. For example, Cobalt Strike is often used to conduct supply chain attacks, where attackers penetrate networks through third-party services or contractors.

Brute Ratel C4, although ranked only 19th, attracts attention due to its high technology and novelty. The discovery of a hacked version of this tool on underground forums in 2024 significantly increased the number of times it was used in criminal campaigns. This is indicative of a trend where attackers are increasingly using modern pentest tools in their attacks. For example, Latrodectus, a well-known cybercriminal group, uses Brute Ratel C4 as a loader for other malicious components, allowing it to effectively bypass detection systems.

* *CPITS 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2025, Kyiv, Ukraine*

* Corresponding author.

† These authors contributed equally.

buchyk@knu.ua (S. Buchyk); nastiash.2003@gmail.com (A. Shabanova); alex8sbu@knu.ua (O. Buchyk); vika.shmatko.24@knu.ua (V. Shmatko); maksym_kotov@ukr.net (M. Kotov)

0000-0003-0892-3494 (S. Buchyk); 0009-0008-4962-569X (A. Shabanova); 0000-0001-7102-2176 (O. Buchyk); 0009-0008-9619-3789 (V. Shmatko); 0000-0003-1153-3198 (M. Kotov)

Despite the decline in the popularity of some remote access trojans (RATs), such as DCRAT (–59%), NjRAT (–33%), and AsyncRAT (–29%), their overall share of botnet activity remains significant (30.45%). RATs allow attackers to gain full control over an infected system, making them very effective in spying campaigns or data theft. For example, Remcos, which has seen a 72% increase, is actively used to steal credentials, including corporate ones. Cybercriminals can send commands to an infected computer to obtain sensitive information, such as passwords or files.

Overall, the slowdown in the growth of botnets is a positive development, but it does not diminish the relevance of the global implementation of more stringent cyber defense measures. Botnets remain difficult to detect, and their activity is constantly adapting to new defense methods, which requires a comprehensive approach to solving problems.

Geolocation of botnet C&Cs, Jul-Dec 2024 (continued)

Top 20 locations of botnet C&Cs

Rank	Country	Jan - Jun 2024	Jul - Dec 2024	% Change
#1	China	2,823	3,535	25%
#2	United States	2,702	2,286	-15%
#3	Russia	1,302	1,125	-14%
#4	Netherlands	737	782	6%
#5	Germany	742	657	-11%
#6	Bulgaria	715	544	-24%
#7	Singapore	332	382	15%
#8	Mexico	497	334	-33%
#9	United Kingdom	286	317	11%
#10	France	386	279	-28%
#11	Sweden	226	275	22%
#12	Finland	135	213	58%
#12	Morocco	-	213	New entry
#14	Argentina	223	172	-23%
#15	Japan	128	136	6%
#16	Canada	144	128	-11%
#17	Vietnam	149	122	-18%
#18	Korea (Rep. of)	107	110	3%
#19	Colombia	135	107	-21%
#20	Spain	-	100	New entry

Figure 1: Dynamics of the number of attacks. Spamhaus Botnet Threat Update Jul.-Dec. 2024 [1]

1.1. Characteristics of cyber attacks using DGA

Domain Generation Algorithms (DGAs) are tools that allow attackers to dynamically create thousands of unique domains to support command-and-control malware servers [2]. These algorithms work based on various approaches, such as arithmetic calculations, hashing, or the use of dictionaries. Their main goal is to avoid blocking and ensure stable communication between infected devices and control servers even in the event of a partial infrastructure blockage [3].

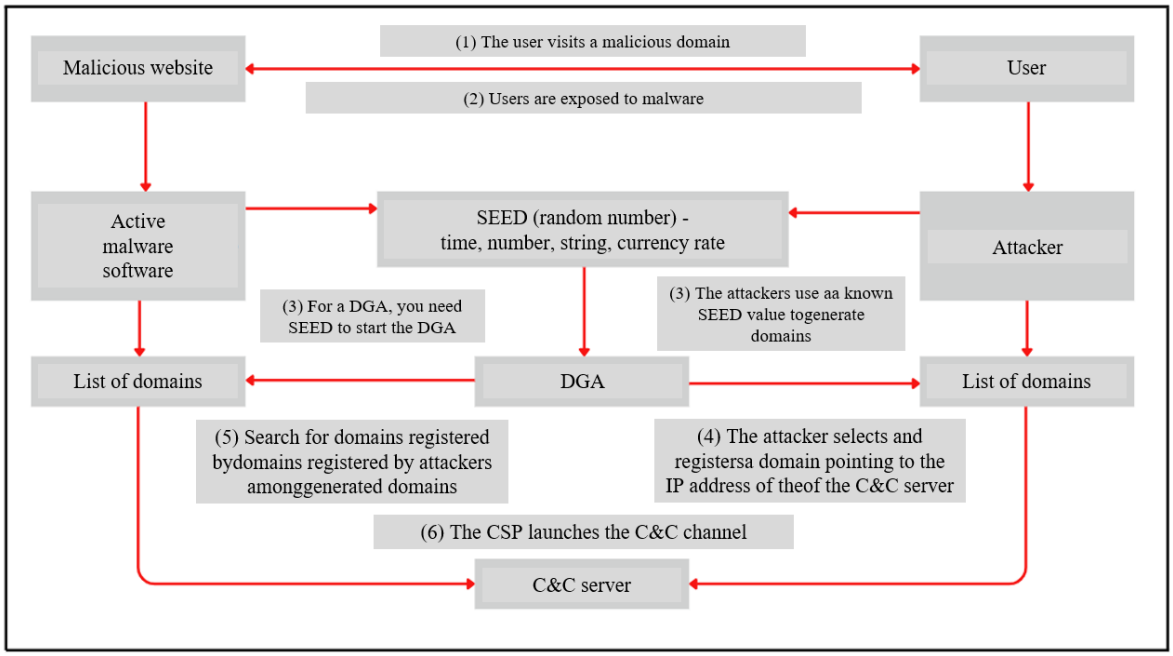

Figure 2: Specifics of using DGA algorithms

One of the main advantages of DGAs is the ability to create a large number of domains in a short time due to the specifics of their formation, as shown in Fig. 2. Attackers can generate thousands of addresses every day, which makes them much harder to detect and block. Moreover, the dynamic nature of generation avoids predictability, making such domains more resistant to traditional detection methods. Another important feature is the minimal dependence on physical infrastructure. Thanks to DGA, even after blocking some servers, attackers can quickly switch to new domains.

DGAs are implemented using various algorithms that can be classified according to the approach to domain creation [4].

One of the most common methods is arithmetic-based algorithms. In this case, domains are generated using mathematical formulas that take into account parameters such as date, time, or other variables. For example, the Conficker virus used this approach to create pseudo-random domains, such as 'fgavropgu.com', which ensured the stability of its command-and-control infrastructure.

Another approach is based on the use of hash functions. In this case, the algorithm generates domains by calculating a hash of input data, such as strings of text or IP addresses [5]. This allows the creation of more complex and less predictable domain names that are difficult to detect. The Bamital virus used this method to create domains, for example, '47faeb4f1b75a48499ba14e9b1cd895a.org', ensuring the high resilience of its infrastructure.

Another great approach is dictionary-based algorithms. In this case, a database of words that make up names is used to generate domains, making them look more natural and less suspicious to detection systems. For example, the Matsnu virus generated domains such as 'catpeakfearinterview.com' using this approach. This technique can significantly reduce the number of false positives from cyber defense systems [6].

1.2. Overview of traditional methods for detecting anomalous domains

DGAs are widely used to organize various types of attacks. In addition to botnets, these algorithms are used in Ransomware, which transmits encryption keys via dynamically generated domains. They are also used in phishing campaigns when the seemingly familiar look of domains helps to deceive users, which makes DGA one of the key tools in the modern arsenal of cybercriminals.

Traditional methods of detecting abnormal domains are based on analyzing static characteristics of domain names and behavioral features of network traffic. One of the most common approaches is the use of blacklists that store known malicious domains. While this method is effective for combating already identified threats, it has significant limitations, as it is unable to respond to new, previously unknown DGA domains. Another approach is to use regular expressions to detect abnormal patterns in domains, such as analyzing domain name length, frequency of character usage, or non-standard structures. However, this method demonstrates limited effectiveness when dealing with modern DGAs, which can create domains with a fairly standardized look, in particular, based on dictionaries.

Behavioral methods are more adaptive to dynamic threats and are based on analyzing network traffic, DNS query frequency, domain lifecycle, and domain relationships. For example, traffic analysis systems can detect anomalous activity if a particular domain receives a significant number of simultaneous requests from many IP addresses. However, these methods also have weaknesses, as they depend on a large amount of historical data and may have a high false positive rate. As a result, traditional detection methods remain effective only for a limited range of tasks and need to be supplemented by modern approaches, including those based on artificial intelligence and machine learning models [7–9].

1.3. Using artificial intelligence to detect dynamic domains

The introduction of artificial intelligence (AI) technologies in cybersecurity has significantly improved the efficiency of detecting dynamically generated domains (DGAs), as AI's ability to automatically process large amounts of data and identify hidden patterns ensures high accuracy in detecting new threats that cannot be identified using static approaches [10].

One of the key areas of AI application is the analysis of the structural characteristics of domain names. In particular, machine learning algorithms allow us to identify such features as the frequency distribution of characters, domain length, morphological features, and the use of special characters. Classification models such as Random Forest or gradient boosting are used to analyze these characteristics. In addition, natural language processing (NLP) methods can detect DGA domains that mimic natural words [11–13]. In addition, domain vectorization using Word2Vec or similar tools is used to find hidden patterns in their structure.

Another important approach is modeling network traffic behavioral patterns. This method is based on the analysis of parameters such as the frequency of DNS queries, the frequency of domain accesses, and the distribution of queries by IP address. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) models allow for analyzing time series of traffic and identifying anomalies that may be related to DGA activity. Additionally, clustering algorithms such as k-means help to group domains by behavioral characteristics, which allows you to identify new potentially malicious groups.

Considerable attention is also paid to deep analysis based on neural networks. Deep learning models, such as autoencoders, are used to reduce the dimensionality of data and search for anomalies in the structure of domains [14]. Generative Adversarial Networks (GANs) allow to creation of synthetic data for model training, in particular, to simulate the behavior of DGA domains. In addition, transformers, which have become popular due to text data processing, are used to analyze complex dependencies in the structure of domains.

The practical implementation of such methods has already found its application in modern cyber defense systems. For example, AI-based tools are integrated into security information and event management systems (SIEM), such as Splunk or IBM QRadar. They allow analysing DNS traffic in real-time and identifying abnormal patterns. Open-source solutions, such as PyDGADetector, use TensorFlow and PyTorch libraries to create customized DGA detection models, making this approach affordable [15].

2. Concept and implementation of the hybrid model

The hybrid model for DGA domain detection combines convolutional neural networks (CNNs), bidirectional LSTMs (BiLSTMs), and a self-focused mechanism to efficiently analyze local and global domain name features, as illustrated in Fig. 3. This architecture allows us to take into account the context and highlight the key features of the data, ensuring high accuracy and performance of the model. This paper aims to develop a hybrid machine-learning model for detecting domains generated by algorithms aimed at compromising information systems or misleading users. The object of the study is the process of detecting malicious domain names created by domain generation algorithms (DGAs), and the subject is machine learning architectures that can analyze and classify domain names for their legitimacy [16].

The scientific novelty of the work lies in the further development of methods for detecting fraudulent domains generated by algorithms through the integration of modern machine learning architectures that provide rapid recognition of potential cyber threats. This approach allows for increasing the level of cybersecurity due to the model's ability to adapt to dynamic changes in the behavior of botnets, which remain the main threat in modern cyberspace.

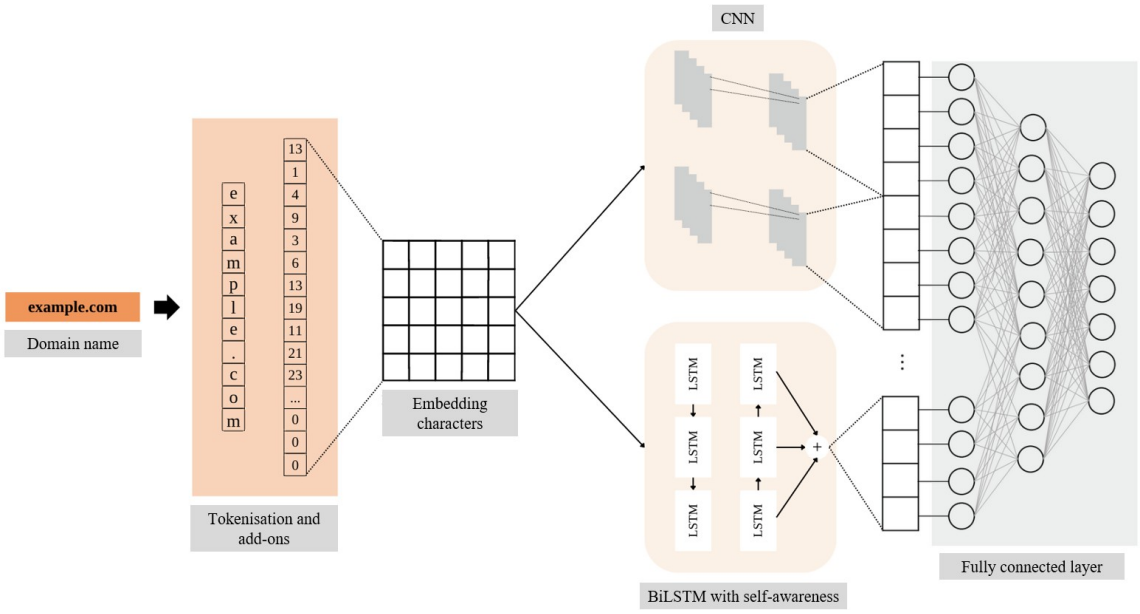

Figure 3: Schematic representation of the hybrid model

2.1. Architectural solutions for the integration of CNN, BiLSTM, and self-attention mechanism

The hybrid model proposed to detect domains generated by DGA algorithms uses a combination of three key components: BiLSTM [17], Self-Attention Mechanism, and Convolutional Neural Networks (CNN) [18]. This architecture provides a multi-level analysis of input data, taking into account local, global, and the most significant features specific to the domains created by the algorithms [19].

BiLSTM is the basis of the model, which allows taking into account the context of characters in a sequence both from left to right and from right to left, which ensures the model's ability to analyze complex patterns in domain names that cannot be analyzed by simple pattern analysis. The use of bidirectional analysis helps to take into account the relationships between characters, which increases the accuracy and reliability of the classification. In addition, BiLSTM's retention and forgetting mechanisms allow you to focus only on the most relevant features, ignoring irrelevant information [20].

The self-focus mechanism adds to the model the ability to highlight key features in domain names. It focuses on the most relevant parts of the data, ignoring secondary elements, which reduces noise and improves classification accuracy to detect domains that include random or artificially generated sequences. The self-awareness mechanism makes the model more resilient to challenging conditions and ensures high performance in real-time data processing [21].

Convolutional Neural Networks (CNNs) perform domain structural analysis by identifying local patterns that are specific to malicious domains. This component of the model extracts information about character frequency, domain name length, and specific sequences, which allows for a better understanding of the structure of the input data. Thanks to dynamic learning, CNNs can adapt to changes in domain patterns and extract important features even from new types of data.

2.2. Data preparation features: tokenization and standardization

The data preparation process is an important step in ensuring model accuracy. Domain names undergo tokenization—breaking them down into individual characters or bigrams, allowing us to preserve the structure and relationships between the elements of the domain name necessary for model analysis. After tokenization, the data is standardized: domain names are leveled to the same length by adding zero values (padding) or trimming redundant characters [22].

Filtering is also performed: characters that are not relevant to the classification, such as rare or special characters, are removed. This process helps to reduce the dimensionality of the problem and reduces the load on the model while preserving the key characteristics of the domains [23]. This approach allows the model to receive clean and standardized data, which is the basis for successful training and accurate prediction.

2.3. Model implementation in TensorFlow and Keras: algorithms and optimization

The model was implemented using TensorFlow and Keras, which provide high flexibility and support for complex architectures. The model architecture is built using a modular approach that allows changing configurations, testing different parameters, and identifying the optimal ones. The model uses convolutional layers with different filter sizes that remove local features and BiLSTM that takes into account contextual dependencies. A self-awareness mechanism is integrated to focus on key sequence features.

The model is trained using the Adam optimizer, which ensures a fast and stable reduction of the loss function. Validation is performed after each epoch, and an early stopping mechanism is used to prevent overfitting. To improve performance and reduce training time, GPU computing is implemented. Thanks to this approach, the model demonstrates high performance and accuracy, which allows it to be effectively used for real-time detection of DGA domains [24].

3. Experimental research

The effectiveness of the proposed hybrid model for domain name classification is determined by comprehensive testing based on key metrics. Evaluation of accuracy, precision, completeness, and F1-indicator reveals the advantages of the chosen architecture in comparison with traditional approaches. This approach provides not only a quantitative characterization of the model's performance but also emphasizes its practical applicability in real-world cyber defense.

The study is aimed at comparing the performance of the hybrid model with current popular approaches to DGA domain detection, including Random Forest, SVM, and other machine learning models. An important aspect is also the analysis of the computational performance and scalability of the model, particularly in the context of its integration into real-time systems. This allows us to determine the model's potential for widespread implementation in automated threat monitoring systems.

3.1. Evaluation of the model's effectiveness by key metrics

The key metrics used to evaluate the model's performance were accuracy, precision, completeness, and F1 indicators. These indicators are necessary to understand the strengths and weaknesses of the model in separating between positive (fraudulent) and negative (legitimate) classes.

They are calculated based on four possible outcomes: true positive (TP); false positive (FP); true negative (TN); and false negative (FN). TP corresponds to cases when the model correctly predicts a positive class. Thus, in the context of detecting fraudulent domains, this result is true when a resource is correctly identified as fraudulent and it is. FPs correspond to cases when the model incorrectly predicts a positive class. In other words, a false positive occurs when a legitimate domain is mistakenly identified as malicious. TNs are results when the model correctly predicts a negative class. A true negative result occurs when a domain is correctly detected as legitimate and is indeed so. FN are results when a negative class is incorrectly predicted. In other words, a false negative occurs when the model cannot identify a fraudulent resource and incorrectly classifies it as legitimate [25].

Accuracy is a fundamental metric that measures the overall correctness of an ML model. It quantifies the ratio of correctly predicted cases (both true positive and true negative) to the total number of instances in the dataset, as defined in formula (1). High accuracy indicates the ability of the model to make correct predictions about positive (fraudulent) and negative (legitimate) cases [26].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

Accuracy measures the correctness of positive predictions made by the model. It quantifies the ratio of true positive predictions to the total number of cases predicted as positive (TP, and FP), as formulated in formula (2). Accuracy is particularly important because it measures the model's ability to avoid misclassifying legitimate domains as malicious. A value of high accuracy means a low frequency of false positives.

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

Completeness, also known as sensitivity or true positive rate, assesses the ability of a model to identify all positive cases in a dataset. It measures the ratio of true positive predictions to the total number of actual positive cases (true positive and false negative), as formulated in equation (3). A high level of completeness is crucial, as it indicates the model's ability to detect a significant proportion of actual online threats while minimizing false negatives.

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

The F1 indicator is the harmonic mean of accuracy and completeness, which provides a balanced assessment of the model's performance, as formulated in the formula (4).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4}$$

This setting is preferred for situations where both high accuracy and completeness are required. F1 is especially necessary when striking a balance between accurately identifying fraudulent websites and minimizing false positives is crucial.

Thus, the use of the described metrics contributes to an objective assessment of the model's performance and its ability to accurately identify classes of input data [27].

The accuracy of the model was 99.25%, which indicates a high ability of the model to correctly classify both legitimate and fraudulent domains. The precision reached 99.26%, indicating a

minimum number of false positive classifications, which is important in the context of ensuring access to legitimate resources. The model's completeness was 99.25%, demonstrating the model's ability to detect real threats without significantly missing fraudulent domains. The F1 score, which takes into account both accuracy and completeness, was 99.25%, highlighting the model's balance and reliability across the different types of domains shown in Fig. 4.

The learning dynamics confirm the effectiveness of the chosen architecture: throughout ten epochs, there was a steady increase in the accuracy and F1-indicator, which approached one. This indicates the model's ability to generalize the acquired knowledge for invisible data. The graphs of training and validation metrics show that the model successfully adapts to the data, minimizing classification errors even when the test data sets are varied [28].

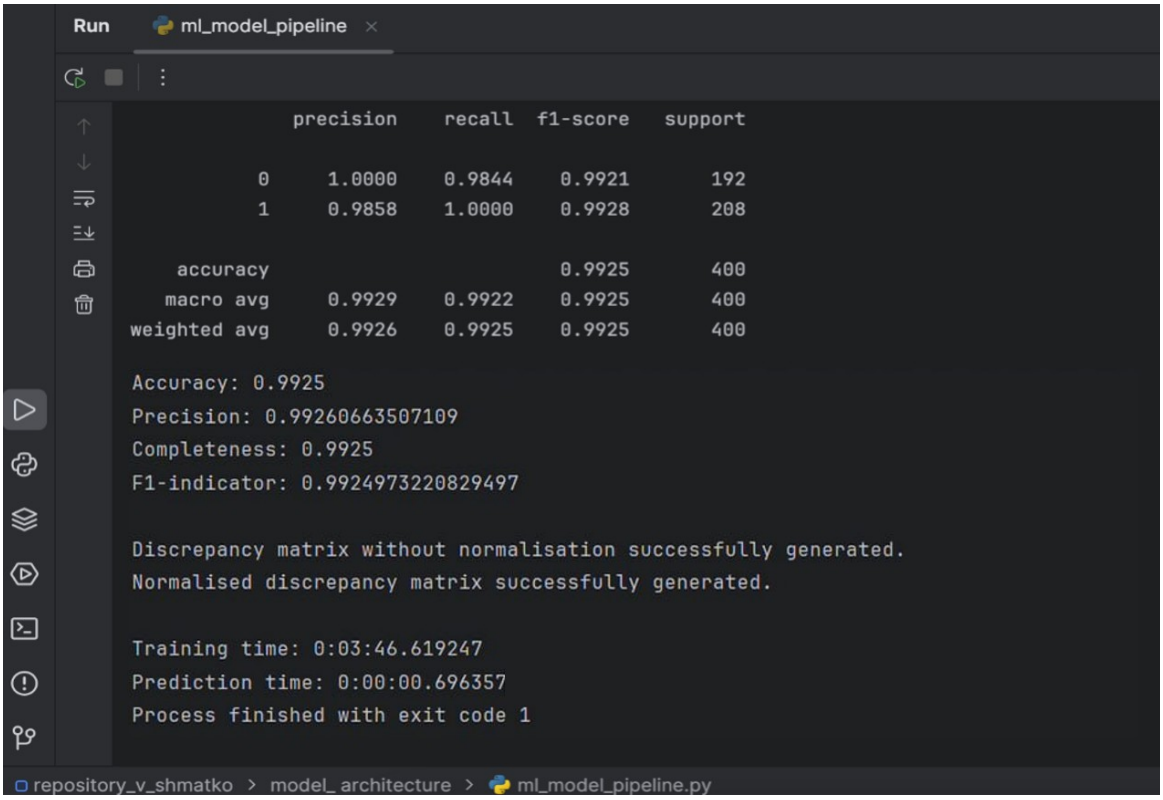

Figure 4: Results of the implemented model. Metrics values.

3.2. Comparison of the hybrid model with popular approaches

The study compared the hybrid model with traditional algorithms such as Random Forest, Support Vector Machines (SVM), and simple metric-based neural networks, as shown in Fig. 5. The hybrid model outperformed these approaches across all key metrics, demonstrating a significant advantage in detecting complex patterns in domain names. For example, compared to Random Forest, the model's accuracy increased by more than 15%, and its precision and completeness exceeded the SVM's results by 10%.

The main advantages of the hybrid model are its ability to integrate local and global data features through a combination of CNN and BiLSTM, as well as a self-focusing mechanism. This allows it to better handle uneven and non-standard domain names, which are typical for DGA. At the same time, the model is more resource-intensive due to the need to use GPUs for optimal training, which may be a limitation for use in low-power environments.

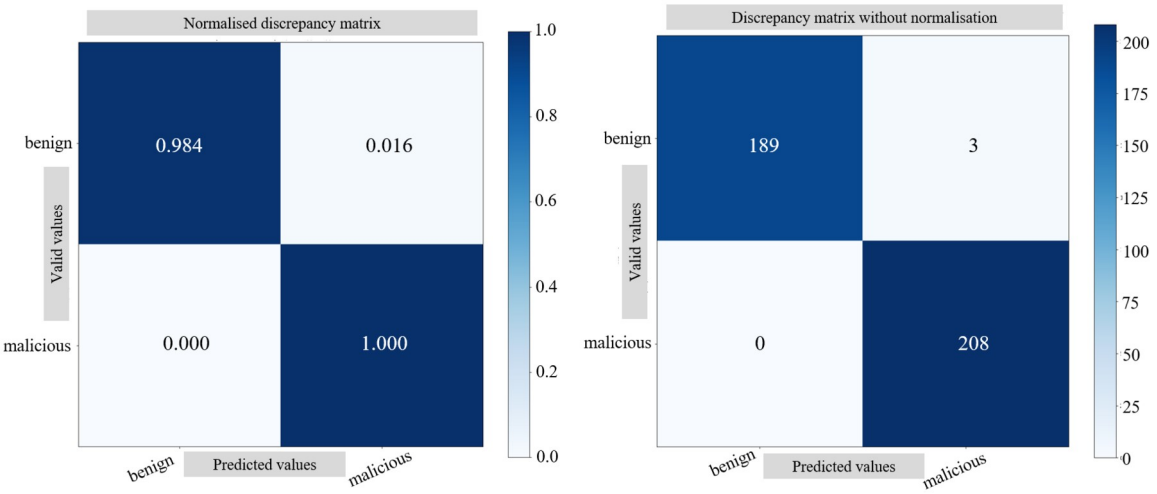

Figure 5: Results of the implemented model. Evaluation of metrics

3.3. Analysis of computational performance and scalability in real conditions

The hybrid model training process was optimized to ensure efficient use of computing resources. The model's training time of 3 minutes and 46 seconds demonstrates its high performance (Fig. 6), even when processing large amounts of data, which allowed the model to achieve high results in a limited time, which is critical for rapid deployment in real-world systems.

Predictions on the test dataset were performed with an average time of 696 milliseconds per prediction, indicating the model's ability to operate in real-time, demonstrating the ability to process large datasets at high speed, making it suitable for enterprise networks and critical infrastructures.

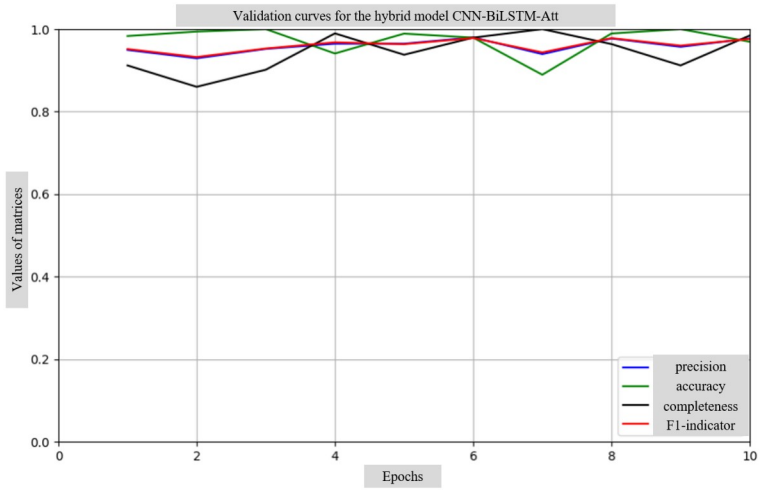

Figure 6: Results of the implemented model. Dynamics of changes in validation metrics for the model in the process of model training

Conclusions

The model code should consist of three modules: the first is aimed at preparing and processing data, the second is responsible for creating and training the model, and the third is focused on evaluating the results.
Central aspects of the model include:

1. Data preparation: the model correctly processes domain names, converting them into tokens at the character level.

2. Structure: a hybrid framework with CNN for feature extraction from text, BiLSTM for context analysis in both directions, and a self-attention mechanism for emphasizing significant characters to improve the model's accuracy.
3. Optimization: the use of the Adam optimizer and mechanisms for early stopping and saving the best model helps to avoid overtraining and provides balanced learning.
4. Evaluation and visualization of results: using a set of metrics such as accuracy, precision, completeness, F1-index, and visualization of data through learning curves and mismatch matrices provides a comprehensive report on the model's performance.

The model demonstrates the potential for highly accurate domain classification, which is key in the fight against targeted cyber threats. Its flexible structure also allows it to be easily adapted to modern security requirements and data types.

The developed solution can be implemented in real-world cybersecurity applications, providing a foundation for further research and expansion in this area. The integration of the developed model is a priority and potentially very useful task given the constant growth of cyber threats. Given that many cyberattacks begin with the disguise of fraudulent domains as legitimate ones, the ability to quickly and accurately identify such resources will significantly reduce the risk of threats being realized.

Implementation of this model in cybersecurity systems such as firewalls, intrusion detection systems, and endpoint security utilities, systems that were presented by the authors in [29], should provide more thorough and reactive protection. In addition, browser extensions with the implemented model will provide an additional layer of link verification.

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] Spamhaus botnet threat update Q4 2024. URL: https://info.spamhaus.com/hubfs/Botnet%20Reports/Jul-Dec%202024%20Botnet%20Threat%20Update.pdf?hsCtaTracking=2d000669-926f-444b-b656-98782e9af734%7C9e5ecf5c-f3a0-4532-b871-bc3213691253

[2] D. Ruts, Improved DGA-based botnet detection through context-related feature selection based on packet flow information, Master's Thesis, 2023.

[3] S. Kapan, S. E. Gunal, Improved phishing attack detection with machine learning: A comprehensive evaluation of classifiers and features, Appl. Sci. 13(24) (2023). doi:10.3390/app132413269

[4] S. Buchyk, et al., DGA domain detection in Splunk with a hybrid machine learning model, in: 2024 IEEE 17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, 2024, 261–264. doi:10.1109/TCSET64720.2024.10755590

[5] S. Sriram, K. P. Soman, M. Alazab, Malicious URL detection using deep learning, Authorea Preprints, 2023.

[6] L. Thammareddi, et al., Analysis On cybersecurity threats in modern banking and machine learning techniques for fraud detection, An Int. Multidisciplinary Online J. (2023).

[7] M. Adamantis, V. Sokolov, P. Skladannyi, Evaluation of state-of-the-art machine learning smart contract vulnerability detection method, Advances in Computer Science for Engineering and Education VII, vol. 242 (2025) 53–65. doi:10.1007/978-3-031-84228-3_5

[8] V. Buhas, et al., Using machine learning techniques to increase the effectiveness of cybersecurity, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3188, no. 2 (2021) 273–281.

[9] V. Zhebka, et al., Methodology for predicting failures in a smart home based on machine learning methods, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 322–332.
[10] G. Andresini, A. Appice, AI meets cybersecurity. J Intell. Inf. Syst. (2023).
[11] O. Romanovskyi, et al., Accuracy improvement of spoken language identification system for close-related languages, Advances in Computer Science for Engineering and Education VII, vol. 242 (2025) 35–52. doi:10.1007/978-3-031-84228-3_4
[12] O. Iosifova, et al., Analysis of automatic speech recognition methods, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 2923 (2021) 252–257.
[13] I. Iosifov, et al., Natural language technology to ensure the safety of speech information, in: Cybersecurity Providing in In-formation and Telecommunication Systems, vol. 3187, no. 1 (2022) 216–226.
[14] S. Das, P. Gangwani, H. Upadhyay, Integration of machine learning with cybersecurity: applications and challenges, in: Artificial Intelligence in Cyber Security: Theories and Applications. Intelligent Systems Reference Library, vol. 240, 2023.
[15] A. S. Saabith, et al., A survey of machine learning techniques for anomaly detection in cybersecurity, Int. J. Res. Eng. Sci. (2023).
[16] 5 Types of LSTM recurrent neural networks, 2023. URL: https://www.exxactcorp.com/blog/Deep-Learning/5-types-of-lstm-recurrent-neural-networks-and-what-to-do-with-them
[17] H. C. Shin, H. R. Roth, M. Gao, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Transact. Medical Imaging 35(5) (2016) 1285–1298. doi:10.1109/TMI.2016.2528162
[18] Z. Alshingiti, et al., A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics, 2023.4 Types of Classification Tasks in Machine Learning, 2020. URL: https://machinelearningmastery.com/types-of-classification-in-machine-learning
[19] A. A. Al Odaini, et al., Cybersecurity in public space: Leveraging CNN and LSTM for proactive multivariate time series classification. In: IEEE International Conference on Big Data, 2023.
[20] H. Lin, et al., A new method for heart rate prediction based on LSTM-BiLSTM-Att, Measurement, 207 (2023).
[21] Unicode security mechanisms for UTS #39, 2023 URL: https://www.unicode.org/Public/security/latest/confusables.txt
[22] D. Plohmann, et al., A comprehensive measurement study of domain generating malware, The 25th USENIX Security Simposium, USENIX Association, 2016.
[23] L. Zhou, et al., Machine learning on big data: Opportunities and challenges, Neurocomputing 237 (2017) 350–361. doi:10.1016/j.neucom.2017.01.026
[24] F. D. Keles, P. M. Wijewardena, C. Hegde, On the computational complexity of self-attention, in: 34th International Conference on Algorithmic Learning Theory, vol. 201, 2023, 597–619.
[25] T. Ahmad, M. N. Aziz, Data preprocessing and feature selection for machine learning intrusion detection systems, ICIC Express Lett, vol. 13(2), 2019, 93–101. doi:10.24507/icicel.13.02.93
[26] M. N. Alam, et al., Phishing attacks detection using machine learning approach, in: 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT), 2020. doi:10.1109/ICSSIT48917.2020.9214225
[27] R. Gupta, et al., Machine learning models for secure data analytics: A taxonomy and threat model, Comput. Commun. 153 (2020) 406–440. doi:10.1016/j.comcom.2020.02.008
[28] G. Logeswari, S. Bose, T. Anitha, An intrusion detection system for SDN using machine learning, Intell. Autom. Soft Comput. 35(1) (2023) 867–880. doi:10.32604/iasc.2023.026769
[29] S. Toliupa, et al., Building an intrusion detection system in critically important information networks with application of data mining methods, in: 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, 2022, 128–133. doi:10.1109/TCSET55632.2022.9767029