# Method for detecting malicious commands transmitted via images using steganography*

Dmytro Denysiuk[1,*,†] , Oleg Savenko[1,†] and Miroslav Kvassay[2,†]

[1] *Khmelnytskyi National University, Instytutska Str., 11, Khmelnytskyi, 29016, Ukraine*
[2] *Zilina University, Univerzitná 8215, 010 26 Žilina, Slovakia*

### Abstract

This article proposes a method for detecting malicious commands transmitted through graphic files using steganography. The method is based on the analysis of the discrete cosine transform (DCT) to assess changes in the frequency spectrum of an image, combined with behavioral analysis of the system after opening a suspicious file. The proposed approach not only enables the detection of hidden data within graphic objects but also classifies steganographic embedding methods, significantly improving the effectiveness of concealed threat detection.

Experimental studies conducted using publicly available datasets, including BOSSBase, StegoAppDB, and the ALASKA Steganalysis Dataset, have confirmed the superiority of the proposed approach over state-of-the-art steganography detection methods such as SRNet, Xu-Net, and Yedroudj-Net. The obtained results demonstrate high accuracy and recall, reaching 93.4% and 90.8%, respectively, in the task of detecting hidden data, as well as 81.7% and 78.2% in the task of classifying the steganographic embedding method. The combination of DCT analysis with system behavioral characteristics has significantly improved detection performance, reducing the number of hidden threats that could be exploited for transmitting malicious commands.

The proposed method has promising applications in cybersecurity for monitoring multimedia content, analyzing information flows, and detecting covert data transmission channels. Future research may focus on enhancing classification algorithms and integrating transformer models for a deeper analysis of frequency variations in graphic files.

### Keywords

steganography, hidden command detection, discrete cosine transform, behavioral analysis, machine learning, information security

## 1. Introduction

In the modern digital environment, information security has become increasingly critical, particularly in the context of threats associated with the covert transmission of malicious commands through graphic files. The use of steganography [1] enables attackers to effectively conceal communication by embedding malicious instructions into digital images without visibly altering their structure. This technology poses a significant threat, especially given the widespread use of multimedia data in web resources, social networks, advertising systems, and corporate networks, where traditional cybersecurity measures often prove insufficient.

Steganographic methods [2] have a long history of use for covert information transmission; however, in the digital era, they have become a powerful tool for cybercriminals. The concealment of commands within graphic files is employed to bypass security control mechanisms, which is particularly relevant for botnets [3], malicious scripts, and attacks aimed at compromising corporate

*Intelitsis'25: The 6th International Workshop on Intelligent Information Technologies & Systems of Information Security, April 04, 2025, Khmelnytskyi, Ukraine*

[*] Corresponding author.

[†] These authors contributed equally.

denysiuk@khmnu.edu.ua(D. Denysiuk); savenko_oleg_st@ukr.net (O. Savenko); miroslav.kvassay@fri.uniza.sk (M. Kvassay)

0000-0002-7345-8341 (D. Denysiuk); 0000-0002-4104-745X (O. Savenko); 0000-0002-6450-5417 (M. Kvassay);

networks. Notably, steganography has been utilized in real-world cyberattacks, such as the Stegano attack [4].

Stegano is an exploit kit discovered in 2016 that utilized steganography to covertly embed malicious code within the pixels of advertising banners on popular websites. This method enabled attackers to infect users' computers without any interaction, merely through visiting compromised websites.

As part of this attack, the attackers modified the transparency (alpha channel) of individual pixels in PNG images used in advertising banners to conceal malicious code. Once these banners were loaded on a web page, the embedded JavaScript code extracted the hidden data and redirected the user to a malicious website. There, the Stegano exploit kit leveraged vulnerabilities in Adobe Flash Player (specifically, CVE-2015-8651, CVE-2016-1019, and CVE-2016-4117) to execute malicious code on the victim's computer.

Another type of attack is the Vawtrak attack [5] (2018). Vawtrak, also known as NeverQuest or Snifula, is a banking Trojan first discovered in 2014. Its primary objective is to steal users' financial data, such as login credentials for online banking. Vawtrak spreads through malicious spam campaigns and downloads of other malware. It is capable of modifying web page content by injecting malicious forms into banking websites to capture users' confidential information. Additionally, Vawtrak employs a domain generation algorithm (DGA) [6] to determine its command-and-control (C2) servers, making it more resistant to detection and blocking.

In 2018, Vawtrak remained an active threat, continuously evolving and refining its attack methods. Notably, researchers identified collaboration between different malware families, including Vawtrak, TrickBot, Gozi, and IcedID. This cooperation enabled cybercriminals to share tools and techniques, significantly enhancing the effectiveness of their attacks.

Moreover, in 2018, Vawtrak employed steganography to conceal malicious components within images, making it more difficult for traditional security measures to detect the malware. This technique allowed attackers to embed malicious code in seemingly harmless files, complicating its identification and mitigation.

One of the primary reasons for the effectiveness of such attacks is the difficulty in detecting hidden commands. Traditional antivirus systems and network traffic analysis tools are not always capable of identifying covert messages, as modifications in graphic files do not alter their visible structure or behavioral characteristics. Moreover, modern steganographic techniques enable the concealment of commands within the frequency components of images, utilizing approaches such as the discrete cosine transform (DCT) [7] or least significant bit (LSB) methods [8].

Modern cybersecurity practices involve the application of machine learning for detecting such threats. In particular, convolutional neural networks (CNNs) [9] have demonstrated high effectiveness in identifying hidden patterns in graphic files that may indicate the presence of malicious command [10]. Additionally, behavioral analysis [11] and anomaly detection methods enable the assessment of changes in network activity, which may signal the use of steganographic communication channels.

Given the relevance of this issue, the primary objective of this study is to develop a method for detecting malicious commands transmitted through images using steganography. The research focuses on formalizing the process of hidden command detection, designing an efficient information system architecture for analyzing graphic files, and implementing machine learning methods to automate detection. The proposed model aims to enhance cybersecurity, minimize the risk of confidential data leakage, and prevent the use of digital images as covert communication channels for malicious actors.

## 2. Literature review

In contemporary research, steganography is considered one of the most effective methods for concealing data, utilized both for maintaining confidentiality and for covertly controlling malicious systems. This section outlines the primary steganographic techniques, common attack strategies leveraging steganography, and the approaches used for their detection.

## 2.1. Modern steganographic methods for command transmission

Steganographic methods used for concealing malicious commands in graphic files can be categorized into three main groups: spatial domain manipulation methods, frequency domain methods, and hybrid approaches.

The least significant bit (LSB) method [12] is one of the most widely used techniques for hiding information in images. It involves replacing the least significant bits of each pixel with the bits of the hidden message. The primary advantage of this method lies in its simplicity of implementation and high data embedding capacity. However, it is vulnerable to statistical analyses, such as intensity histograms, as well as detection through image entropy analysis.

The discrete cosine transform (DCT) method [13] is used for hiding information in the frequency domain of an image. This method serves as the foundation for many algorithms, including JPEG steganography. It offers high resistance to basic statistical attacks; however, it remains vulnerable to more advanced spectral analyses and machine learning techniques capable of detecting anomalies in frequency components.

The discrete wavelet transform (DWT) method [14] enables the concealment of information across different frequency subbands, providing high resistance to noise and image compression. Despite its effectiveness, DWT requires more complex computations and must be adapted to specific graphic file formats, which complicates its practical application in real-world attacks.

## 2.2. The use of machine learning and behavioral analysis for detecting hidden commands

Traditional steganography detection methods, such as statistical histogram analysis [15] and spectral image analysis [16], exhibit limited effectiveness against advanced hiding techniques. In response to these challenges, researchers increasingly employ machine learning algorithms and behavioral analysis to enhance detection capabilities.

In particular, convolutional neural networks (CNNs) are used for the automatic analysis of images to detect hidden patterns. For example, research studies have employed models such as ResNet [17] and EfficientNet [18] for steganographic modification detection. The advantage of CNNs lies in their high recognition accuracy; however, their main drawbacks include the need for large training datasets and high computational complexity.

Recurrent neural networks (RNNs) [19] and transformers are also being explored as tools for analyzing sequential changes in images, particularly in video files. However, these approaches have not yet seen widespread adoption in practical cybersecurity systems.

Behavioral analysis of network [20] traffic enables the detection of anomalies in system communications that may indicate covert command transmission [21]. Anomaly-based detection methods can identify atypical interaction patterns between clients and servers; however, they are highly sensitive to false positives, which can impact their reliability in practical applications.

## 2.3. The need for developing new detection process models

Despite significant progress in the detection of steganographic threats, existing methods have certain limitations. Traditional statistical approaches exhibit low effectiveness against advanced hiding algorithms, while machine learning methods require large volumes of training data and substantial computational resources. Additionally, modern attacks can employ adaptive techniques, dynamically altering their steganographic methods to evade detection.

Thus, there is a need to develop a new detection process [22] model for identifying malicious commands transmitted through images using steganography. This model should integrate in-depth analysis of graphic files with behavioral methods [23], enhancing detection accuracy and reducing the risk of covert cyberattacks. The development of a comprehensive approach that combines machine learning techniques, anomaly detection, and cryptographic analysis will be a crucial step in strengthening cybersecurity.

## 3. Detection process model for malicious commands

In the process of detecting malicious commands transmitted through images using steganography, the key components include the command source, the information carrier, and the command decoding mechanism. The command source is a critical element, as it determines the method of generating and transmitting hidden instructions. Typically, this source is a malicious system or command-and-control (C2) server that creates specially encoded commands for compromised devices. These commands can be embedded as bit-level modifications in an image, which is then transmitted through open communication channels such as social networks, websites, or email.

The content of such commands may include instructions for downloading additional malware, modifying system configurations, or initiating attacks on other resources. Since these methods exploit legitimate digital platforms to transmit malicious instructions, detecting such threats requires advanced analysis algorithms and anomaly detection techniques tailored for graphical files.

The information carrier, i.e., the digital graphic file, serves as the primary medium for concealing commands. The embedding of hidden information is performed using steganographic methods, with the most commonly employed techniques being the least significant bit (LSB) method, discrete cosine transform (DCT), and discrete wavelet transform (DWT). The choice of a specific method depends on the required level of concealment, the type of graphic file, and the resistance to detection.

For instance, the LSB method is effective for making minor modifications to high-quality images, ensuring that the changes remain imperceptible to the human eye. Meanwhile, DCT and DWT methods allow hidden data to be integrated into the frequency components of the file, offering a higher level of protection against visual analysis and automated detection techniques.

The command decoding mechanism operates on the recipient’s side, which may be a compromised device or a specialized software platform that extracts the hidden information and executes the received instructions. The decoding process is carried out using predefined steganographic keys or machine learning algorithms that automatically identify concealed data.

In more sophisticated cases, attackers employ adaptive recognition mechanisms that analyze the contextual content of images, allowing them to embed additional information more effectively while making detection even more challenging. For this reason, traditional analysis methods are not always sufficiently effective, necessitating hybrid approaches that combine multiple detection techniques simultaneously.

The transmission of malicious commands through images occurs in multiple stages, including command generation and encryption, embedding into a graphic file using steganographic algorithms, transferring the modified image via open communication channels, and subsequently extracting and executing the command on the recipient's side. Encrypting the command before embedding it in the image enhances its protection against detection and unauthorized access.

The use of various transmission channels alters traditional approaches to cyber threat analysis, as attackers exploit publicly available resources such as social networks, file storage services, or web platforms, which are not subject to rigorous monitoring by security systems. On the recipient’s side, the extraction of hidden information is executed according to predefined decoding algorithms. If the detection process is successful, the compromised system automatically executes the received commands, potentially leading to further system infection or its involvement in large-scale attacks.

To complicate analysis and detection, attackers employ dynamic command reconstruction, adapting the extracted instructions to the specific execution environment.

The detection of hidden commands in graphic files requires a comprehensive analysis that incorporates statistical, behavioral, and neural network-based methods. Statistical analysis serves as the initial stage of the investigation, aimed at identifying atypical alterations in the image structure. Entropy analysis helps assess the level of randomness in the data, which may indicate the presence of concealed information. Additionally, a color histogram analysis is performed, as abrupt changes in the distribution of shades may result from steganographic operations. Pixel structure analysis further enables the identification of modifications in the least significant bits, which are characteristic of steganographic embedding techniques.

The next step involves analyzing the behavioral characteristics of the system after loading the image. Unusual network activity, increased CPU usage, or the execution of suspicious processes may indicate the presence of hidden commands. Behavioral analysis is an effective approach for detecting anomalous patterns in system operation, which may suggest the execution of malicious code. The use of machine learning to model normal system behavior enables the identification of anomalies that could indicate unauthorized actions.

The final stage involves the application of deep neural networks for the automatic classification of suspicious images. Convolutional neural networks (CNNs) effectively detect subtle pixel structure modifications that may indicate the presence of hidden information. Recurrent neural networks (RNNs) are used to analyze sequential changes in images, making them particularly useful for examining video files. Transformer models analyze complex correlations between different parts of an image, enhancing the detection efficiency of steganographic manipulations.

The integration of these methods provides a comprehensive approach to detecting malicious commands, significantly improving system protection against cyber threats. Future research may focus on combining classical image analysis techniques with deep neural networks, facilitating the development of even more accurate and resilient models for detecting hidden threats.

## 4. Method for detecting malicious commands in graphic objects

Based on the proposed model, a method for detecting malicious commands in graphic objects has been developed. This method is based on the analysis of the discrete cosine transform (DCT) and the behavioral characteristics of the system after loading the image. The primary objective of the method is to determine the presence of hidden information in a graphic file and subsequently identify the steganographic embedding technique used.

The core approach involves analyzing the spectral characteristics [24] of the image, as modifications in the pixel domain may be too subtle to be detected by traditional methods. Additionally, behavioral analysis enables the identification of signs of hidden command execution after opening the graphic file.

A graphical image can be represented as a matrix of pixel intensities $I(x, y)$, where $(x, y)$, are the pixel coordinates, and each element's value, ranging from 0 to 255, determines the brightness level. In the case of steganographic methods based on DCT, hidden data are embedded into the frequency coefficients, leading to alterations in the spectral representation of the image [25]. To assess these changes, the image is divided into blocks of 8×8 pixels, and the DCT is computed for each block according to the equation:

$$DCT(u,v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} I(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \tag{1}$$

the normalization coefficients are defined as:

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, & u = 0 \\ 1, & u > 0 \end{cases} \qquad C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & v = 0 \\ 1, & v > 0 \end{cases} \tag{2}$$

These coefficients ensure the proper scaling of frequency components, preserving energy distribution during the transformation.

The transformed image in the frequency domain contains low-frequency coefficients [26], which preserve the primary structural information of the image, and high-frequency coefficients, which store finer details that are less perceptible to the human eye. These high-frequency components are typically modified during steganographic data embedding.

To detect hidden information, it is necessary to analyze the statistical characteristics of these coefficients. This includes evaluating their distribution, variance, and entropy to identify anomalies indicative of steganographic alterations.

Changes in the frequency spectrum are detected by computing the mean and variance of the high-frequency coefficients. The mean value is calculated as:

$$\mu_{HF} = \frac{1}{N}\sum_{i=1}^{N} DCT_i^{HF} \tag{3}$$

where $DCT_i^{HF}$ – represents the high-frequency DCT coefficients, a $N$ – is the total number of such coefficients in the analyzed image blocks. This measure provides an estimate of the overall intensity of modifications in the frequency domain.

The variance is calculated by the formula:

$$\sigma_{HF}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(DCT_i^{HF} - \mu_{HF}\right)^2 \tag{4}$$

To assess the deviation of these parameters from normal values, the statistical anomaly criterion is used:

$$Z = \frac{|\mu_{HF} - \mu_0\,|}{\sigma_0} \tag{5}$$

where $\mu_0$ i $\sigma_0$ – mean and standard deviation for a large sample of images without steganography. If $Z$ exceeds a certain threshold $Z_{threshold}$, a conclusion can be made about the possible use of steganography.

After confirming the presence of steganographic changes, it is necessary to determine which method of concealment was used. To do this, a feature vector is formed that contains the skewness coefficient of the distribution of DCT coefficients:

$$S = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{DCT_i - \mu}{\sigma}\right)^3 \tag{6}$$

kurtosis ratio (assessment of the peak distribution):

$$K = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{DCT_i - \mu}{\sigma}\right)^4 - 3 \tag{7}$$

and entropy of high-frequency components:

$$K = -\sum_{i} p_i \log p_i \tag{8}$$

where $p_i$ – the probability of a specific value of the DCT coefficient. Based on these characteristics, a machine learning model is used, specifically an LSTM network [27], which is trained on a dataset of images with known hiding methods, allowing it to identify the applied steganographic method.

In addition to analyzing frequency characteristics, behavioral analysis of the system is performed after opening the graphic file to detect potential malicious commands. If the image contains hidden commands, it may trigger certain malicious actions, which can be identified by changes in system process activity. Let$P(t)$ – is the vector of active processes at time $t$. The activity deviation is defined as:

$$D = \sum_{i} |P_i(t + \Delta t) - P_i(t)| \tag{9}$$

where $\Delta t$ – time interval after opening the file. If the value $D$ exceeds the permissible threshold value $D_{threshold}$, this may indicate the execution of hidden code.

The evaluation of the method's effectiveness [28] is conducted using standard accuracy and recall metrics. Accuracy is defined by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

where $TP$ – number of correctly detected images with steganography, $TN$ – number of correctly classified normal images, $FP$– number of false alarms, $FN$ – number of missed steganographic files. Completeness is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

that allows to evaluate the ability of the method to correctly detect all cases of steganography. The proposed method combines the analysis of image frequency characteristics, statistical estimates, and behavioral monitoring, which significantly improves the accuracy of steganographic attack detection. The integration of these approaches provides a comprehensive cybersecurity mechanism capable of detecting hidden threats even in cases of complex steganography methods

## 5. Experiments

To evaluate the effectiveness of the proposed method for detecting malicious commands transmitted through steganography in graphic files, experimental studies were conducted. The primary objective was to assess the accuracy of detecting hidden information in images, identify the steganographic embedding method, and evaluate the effectiveness of behavioral analysis after opening the graphic file. For this purpose, large datasets were used, containing both clean images and images modified using various steganographic techniques.

One of the primary datasets used for the research was BOSSBase [29]. It contains over 10,000 high-quality grayscale images, widely utilized in steganographic studies. A key feature of this dataset is that all images have a uniform size and lack visible artifacts, allowing for the minimization of external factors' influence on the analysis results. Based on this dataset, synthetic data were generated by embedding hidden messages using various steganographic methods.

The second dataset used was StegoAppDB [30], which contains images modified using various mobile steganography applications. The uniqueness of this dataset lies in the fact that it includes data obtained from real devices, accounting for variations related to different camera parameters and compression methods. This makes testing on this dataset more representative of real-world scenarios where steganography is applied in practical use cases.

Another dataset used was the ALASKA Steganalysis Dataset [31], which is one of the largest and most representative datasets for steganography detection. It contains JPEG-format images modified using various hiding methods, including DCT and LSB steganography. The use of this dataset allows for evaluating the effectiveness of the proposed method in cases where hidden data are embedded in frequency domains, making it particularly relevant for testing the proposed approach.

Before the experiments, all images were converted to the JPEG format, as the DCT method is predominantly applied to this type of file. Subsequently, image normalization was performed, which involved resizing all files to **512×512** pixels to eliminate potential effects of varying resolutions on the analysis results. Additionally, metadata cleaning was conducted to prevent the detection of steganographic modifications through meta-information analysis.

The tools used to create the test data were Steghide, F5, Jsteg, OutGuess, and JPEG Steganography Suite. Each of these methods has its own specifics of image modification. The Steghide method uses data encryption and compression algorithms before embedding, which makes it difficult to detect by traditional analysis methods. The F5 method applies adaptive compression, changing the DCT coefficients but minimizing the change in file size, which makes it difficult to detect. Other methods, such as Jsteg and OutGuess, use modification of the least significant DCT coefficients, which allows hiding information with minimal distortion of the visual representation of the image.

The results of the experiments are shown in the table, which demonstrates the accuracy and completeness for the task of detecting steganography and classifying the hiding method.

**Table 1**

Experimental results

| Method detection | Accuracy ( detection ) | Recall ( detection ) | Accuracy ( classification ) | Recall ( classification ) |
|---|---|---|---|---|
| SRNet (SOTA) | 89.3% | 85.6% | 77.4% | 72.8% |
| Xu-Net | 87.2% | 82.1% | 74.3% | 68.5% |
| Yedroudj-Net | 85.8% | 80.2% | 72.5% | 66.3% |
| Requisitions method | 93.4% | 90.8% | 81.7% | 78.2% |

The obtained results show that the proposed method has higher accuracy and completeness compared to state-of-the-art approaches. In particular, the proposed method demonstrates improved Recall [32], which indicates the ability to find more suspicious images than other methods. This is explained by combining the analysis of DCT frequency characteristics with behavioral analysis of changes in the system after opening a file.

Figure 1 shows a graph comparing the Accuracy [33] of different methods in the tasks of detecting hidden information and classifying a steganographic embedding method. It can be seen that the proposed method has the highest accuracy among all tested models, reaching 93.4% in steganography detection and 81.7% in method classification. This indicates that the use of DCT analysis together with behavioral characteristics can significantly improve the detection of hidden commands in graphic files.

Figure 2 shows a graph comparing the recall of different methods in the tasks of steganography detection and classification. The proposed method demonstrates a significant advantage over state-of-the-art approaches, in particular, reaching 90.8% in detecting hidden information and 78.2% in classifying the embedding method. This means that the proposed algorithm correctly identifies a much larger number of images with hidden data, reducing the likelihood of false negatives.

Additional testing of behavioral analysis showed that in 92.1% of cases, the system detected suspicious activity after opening an image that contained hidden commands. In 7.9% of cases, such images did not cause any action, which indicates the limitations of the behavioral approach in cases where hidden commands do not affect the activity of processes in the system.

The obtained results confirm that the application of a combined approach based on the analysis of DCT and system behavioral characteristics can significantly improve the efficiency of detecting steganographic manipulations in graphic files. This is especially important for detecting malicious commands, since traditional image analysis methods do not provide high enough efficiency in cases where hidden data is embedded in frequency domains.

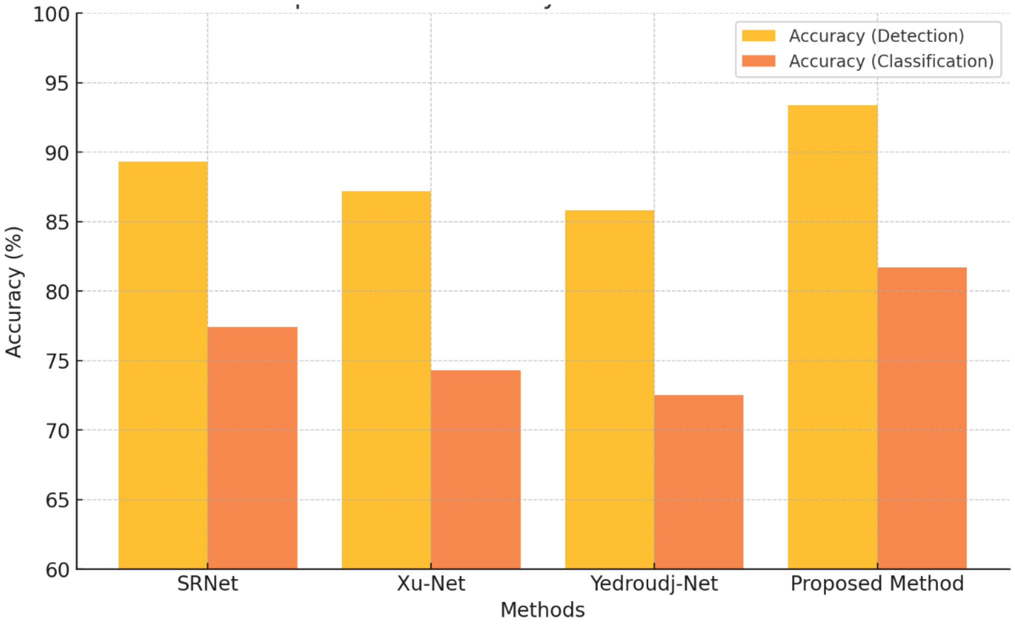

**Figure 1:** Comparison of accuracy for different method.

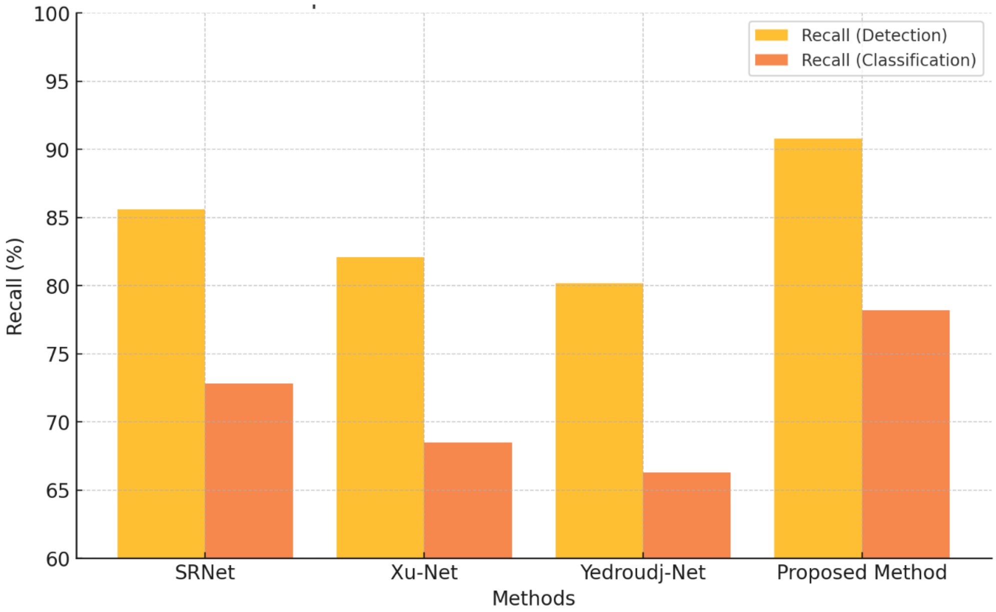

**Figure 2:** Comparison of recall for different methods.

## 6. Conclusions

The article presents a method for detecting malicious commands in graphic files using discrete cosine transform (DCT) analysis and system behavioral monitoring. The approach not only detects hidden information but also classifies steganographic embedding methods, making it effective for real-world cybersecurity applications.

Experimental results show that the proposed method outperforms SRNet, Xu-Net, and Yedroudj-Net, achieving 93.4% accuracy in detecting hidden data and 81.7% accuracy in steganographic method classification. Recall analysis confirms its effectiveness, with 90.8% correctly detected steganographic images and 78.2% correctly classified embedding techniques.

A key advantage is combining frequency analysis with behavioral monitoring, allowing the detection of DCT coefficient changes and malicious actions triggered after opening a file. This reduces hidden threats used for cyberattacks via steganographic channels. Additionally, the method enhances cybersecurity by identifying steganographic techniques that exploit legitimate digital platforms to covertly transmit malicious instructions.

Despite its effectiveness, the method has certain limitations. Behavioral analysis is less effective when hidden data are used passively, as it relies on system activity changes. Additionally, high-quality images sometimes lead to higher false positives, requiring further classifier optimization. Expanding the dataset and refining detection models can help mitigate these issues and improve the overall robustness of the method.

Future research should focus on deeper frequency analysis using transformable architectures and integrating metadata analysis to detect emerging steganographic techniques. Another promising direction is the development of hybrid models that combine traditional statistical methods, machine learning, and cryptographic analysis for more accurate threat detection. The proposed method has practical cybersecurity applications in multimedia analysis, monitoring information flows, and detecting covert data transmission channels used for cyber threats.

## Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check; DeepL Translate in order to: some phrases translation into English; ChatGPT in order to: conduct experiments as a prompt-based tool for creating automated descriptions of art objects. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] S. Rahman, J. Uddin, M. Zakarya, H. Hussain, A. A. Khan, A. Ahmed, M. Haleem, A comprehensive study of digital image steganographic techniques, IEEE Access 11 (2023) 6770–6791.

[2] O. Kuznetsov, E. Frontoni, K. Chernov, Beyond traditional steganography: enhancing security and performance with spread spectrum image steganography, Appl. Intell. 54 (2024) 5253–5277.

[3] S. Lysenko, K. Bobrovnikova, O. Savenko, A botnet detection approach based on the clonal selection algorithm, in: Proc. 2018 IEEE 9th Int. Conf. Dependable Systems, Services and Technologies (DeSSerT-2018, Kyiv, Ukraine, May 24–27, 2018), pp. 424–428.

[4] Stegano exploit kit, 2016. URL: https://www.welivesecurity.com/2016/12/06/stegano-exploit-kit/.

[5] MS-ISAC Cyber Crime Technical Desk Reference, 2018. URL: https://www.cisecurity.org/wp-content/uploads/2018/09/MS-ISAC-Cyber-Crime-Technical-Desk-Reference.pdf.

[6] K. Hu, M. Wang, X. Ma, J. Chen, X. Wang, X. Wang, Learning-based image steganography and watermarking: a survey, Expert Syst. Appl. (2024) 123715.

[7] S. Kaur, et al., A systematic review of computational image steganography approaches, Arch. Comput. Methods Eng. 29 (2022) 4775–4797.

[8] M. Garg, J. S. Ubhi, A. K. Aggarwal, Neural style transfer for image steganography and destylization with supervised image-to-image translation, Multimed. Tools Appl. 82 (2023) 6271–6288.

[9] B. Lindemann, B. Maschler, N. Sahlab, M. Weyrich, A survey on anomaly detection for technical systems using LSTM networks, Comput. Ind. 131 (2021) 103498.

[10] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic, in: Communications in Computer and Information Science 370 (2013) 243-254.

[11] N. Doukas, P. Stavroulakis, N. Bardis, Review of artificial intelligence cyber threat assessment techniques for increased system survivability, in: Malware Analysis Using Artificial Intelligence

and Deep Learning, Springer International Publishing, 2021, pp. 207–222. doi:10.1007/978-3-030-62582-5_7.

[12] K. F. Rafat, S. M. Sajjad, Advancing reversible LSB steganography: addressing imperfections and embracing pioneering techniques for enhanced security, IEEE Access (2024).

[13] W. Chen, et al., Invisible backdoor attack with attention and steganography, Comput. Vis. Image Underst. 249 (2024) 104208.

[14] G. K. Murthy, T. Kanimozhi, Methodologies in steganography and cryptography – review, in: Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough, vol. 4, 2024, pp. 205–214.

[15] P. C. Mandal, et al., Digital image steganography: a literature survey, Inf. Sci. 609 (2022) 1451–1488.

[16] G. M. Makrakis, et al., Industrial and critical infrastructure security: technical analysis of real-life security incidents, IEEE Access 9 (2021) 165295–165325.

[17] Z. Li, et al., A survey of convolutional neural networks: analysis, applications, and prospects, IEEE Trans. Neural Netw. Learn. Syst. 33 (2021) 6999–7019.

[18] E. Avots, A. A. Jafari, C. Ozcinar, G. Anbarjafari, Alzheimer's Disease Neuroimaging Initiative, Comparative efficacy of histogram-based local descriptors and CNNs in the MRI-based multidimensional feature space for the differential diagnosis of Alzheimer's disease: a computational neuroimaging approach, Signal Image Video Process. 18 (2024) 2709–2721.

[19] I. Priyadarshini, C. Cotton, A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis, J. Supercomput. 77 (2021) 13911–13932.

[20] I. Obeidat, M. AlZubi, Developing a faster pattern matching algorithm for intrusion detection systems, Int. J. Comput. 18 (3) (2019) 278–284. doi:10.47839/ijc.18.3.1520.

[21] S. Lysenko, O. Savenko, K. Bobrovnikova, A. Kryshchuk, Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks, Commun. Comput. Inf. Sci. 860 (2018) 385–401.

[22] O. Kehret, A. Walz, A. Sikora, Integration of hardware security modules into a deeply embedded TLS stack, Int. J. Comput. 15 (1) (2016) 22–30. doi:10.47839/ijc.15.1.827.

[23] X. Yang, J. Yuan, H. Yang, Y. Kong, H. Zhang, J. Zhao, A highly interactive honeypot-based approach to network threat management, Future Internet 15 (2023) 127.

[24] M. A. R. Putra, T. Ahmad, D. P. Hostiadi, B-CAT: a model for detecting botnet attacks using deep attack behavior analysis on network traffic flows, J. Big Data 11 (2024) 49.

[25] J. Yin, et al., Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning, World Wide Web (2022) 1–23.

[26] S. Lysenko, BotGRABBER: SVM-based self-adaptive system for the network resilience against the botnets' cyberattacks, in: Commun. Comput. Inf. Sci. 1039 (2019) 127–143.

[27] M. Almehdhar, A. Albaseer, M. A. Khan, M. Abdallah, H. Menouar, S. Al-Kuwari, A. Al-Fuqaha, Deep learning in the fast lane: a survey on advanced intrusion detection systems for intelligent vehicle networks, IEEE Open J. Veh. Technol. (2024).

[28] O. Savenko, S. Lysenko, A. Kryschuk, Multi-agent based approach of botnet detection in computer systems. In: Communications in Computer and Information Science 291 (2012) 171–180. https://doi.org/10.1007/978-3-642-31217-5_19.

[29] BossBase, 2022. URL: https://bossbase.com/.

[30] StegoAppDB, 2022. URL: https://forensicstats.org/stegoappdb/.

[31] ALASKA2 Image Steganalysis, 2022. URL: https://www.kaggle.com/competitions/alaska2-image-steganalysis.

[32] Z. Zhou, Y. Su, J. Li, K. Yu, Q. J. Wu, Z. Fu, Y. Shi, Secret-to-image reversible transformation for generative steganography, IEEE Trans. Dependable Secure Comput. 20 (2022) 4118–4134.

[33] N. Kayhan, S. Fekri-Ershad, Content-based image retrieval based on weighted fusion of texture and color features derived from modified local binary patterns and local neighborhood difference patterns, Multimed. Tools Appl. 80 (2021) 32763–32790.